Universidad
Carlos III de Madrid
www.uc3m.es

SISTEMA DE DETECCIÓN DE PERSONAS EN SECUENCIAS DE VIDEO

Autor: María Alejandra Benavides Blanco

Titulación: Grado en Ingeniería de Sistemas Audiovisuales

Tutor: Julio Villena Román

Fecha: Junio 2014

Una imagen vale más que mil palabras.
Proverbio chino.

Resumen

La detección de objetos en secuencias vídeo es una técnica que nace a raíz de la evolución de la detección de objetos estáticos en imágenes.

Para que un programa sea capaz de detectar en una imagen el objeto en estudio sólo se tiene que extraer sus características determinísticas, por ejemplo, en el caso de querer detectar pelotas de tenis habría que fijarse en su forma redonda y en su color amarillo.

Al poder segmentar un vídeo en una secuencia de imágenes o frames, se pueden aplicar los algoritmos de detección de objetos en cada una de las imágenes que lo constituyen teniendo el objeto localizado a lo largo del vídeo.

El objetivo principal de este trabajo final de grado es extrapolar la detección de personas en imágenes al campo de las secuencias de vídeo. Para ello se ha llevado a cabo el diseño y la implementación de un conjunto de módulos capaces de marcar en qué región de cada *frame* del vídeo se encuentra la persona de interés.

Una vez desarrollados todos los módulos se ha efectuado una evaluación exhaustiva realizando una serie de experimentos en diferentes condiciones para calcular la precisión del sistema.

Palabras clave: detección, seguimiento, vídeo, personas, imagen.

Abstract

Tracking objects in video sequences is the evolution of objects detection, which is the process of finding instances of real world objects such as faces, balls and buildings in images or video.

For a program to be able to detect an object, it has only to extract its deterministic characteristic. For example, in case you want to detect tennis balls, the system would have to focus on the round shape and the yellow colour.

A video can be segmented into a sequence of frames, so the detection's algorithms can be applied in each of the images having the object located along the video.

The main objective of this work is to extrapolate the detection of people in pictures to the field of sequences. To do this, it has been carried out the design and the implementation of capable modules to mark in which region of each frame is located the person of interest.)

Once developed all modules, it has been carried out an intensive evaluation with a series of experiments under different conditions to calculate the accuracy of the system.

Key words: detection, tracking, people, video, image.

Índice

Índice de figuras

Índice de tablas

Índice de ecuaciones

Capítulo 1
Introducción

Capítulo 1

INTRODUCCIÓN

1.1. MOTIVACIÓN

El uso de algoritmos que realicen seguimiento de personas en vídeos, y por lo tanto el paso primero que es la detección de las mismas, se ha visto incrementado en los últimos años debido a que con él se pueden llevar a cabo numerosas actividades como la video-vigilancia, la clasificación de actividades físicas, el análisis de eventos deportivos, la animación…

El problema surge cuando el aspecto de la persona en el plano de la imagen dificulta su detección. Éste se puede ver afectado por alguno de los factores que se explican a continuación:

- El cambio de posición: Las personas, al ser objetos móviles, pueden realizar cualquier movimiento de forma imprevista y cambiar su aspecto de un plano para otro. Por ejemplo, en el caso de los programas de noticias el presentador en un primer momento puede estar hablando de frente a la cámara pero acto seguido puede situarse de perfil para interactuar con otra persona.

- El cambio en la iluminación: La mínima variación de la luz, ya sea en su intensidad, dirección o color, modifica el aspecto de la persona en estudio. Cualquier pequeño movimiento que la persona realiza cambia la iluminación.

- Ruido: Tanto la captación de imágenes como el procesamiento de las mismas introduce un cierto nivel de ruido que afecta a la fiabilidad del algoritmo.

- Oclusiones: La persona de interés puede no ser reconocido por nuestro algoritmo si se encuentra parcialmente tapado por otros elementos que estén en el plano de la imagen.

En este trabajo se pretende implementar un algoritmo de detección de personas en secuencias de vídeo que se vea afectado lo mínimo posible por los factores descritos anteriormente, cuya tasa de error es encuentre entre unos valores aceptables.

A partir del proceso de detección se podrá realizar un seguimiento de la persona en estudio. Este procedimiento se lleva a cabo con el fin de emplearse en un futuro en aplicaciones de seguimiento en tiempo real.

1.2. OBJETIVOS

A continuación se enumeran de forma general los objetivos que se intentan cumplir con la realización de este trabajo final de grado:

- Realizar el estado del arte sobre las técnicas de detección y seguimiento de personas en secuencias de vídeo.

- Diseñar y desarrollar un algoritmo capaz de detectar personas, que en un futuro sirva para el seguimiento de personas en secuencias de vídeos en tiempo real.

- Realizar un análisis de los resultados obtenidos y evaluarlos, junto con el presupuesto, para valorar la viabilidad de implementar este sistema de acuerdo al entorno socio-económico actual.

1.3. ESTRUCTURA DE LA MEMORIA

El contenido de la presente memoria está dividido en 6 capítulos. Seguidamente se explican de forma concisa cada uno de ellos.

- Capítulo 1 – Introducción
    - El documento comienza con un repaso general al contexto de la detección de objetos o personas en imágenes o vídeos. También se explican una serie de objetivos que se deben lograr con el desarrollo del proyecto y la estructura de la memoria.
- Capítulo 2 – Estado del arte
    - Este capítulo se centra en el estado del arte de la detección de personas. Comienza con una breve descripción de la situación actual, para seguir con un análisis de las técnicas y soluciones existentes hasta el momento.
- Capítulo 3 – Diseño e implementación
    - El capítulo 3 contiene el diseño de todos los bloques que en conjunto forman el algoritmo final. Asimismo se explica la implementación de cada bloque, los problemas encontrados en el camino y la solución para cada uno de ellos.
- Capítulo 4 – Evaluación del sistema.
    - Una vez explicados los módulos del algoritmo, el capítulo 4 recoge las diferentes pruebas a las que se sometió nuestro proyecto para poder determinar la eficiencia del mismo. Para entender los resultados se ha incluido un corpus de los vídeos que se han usado para la realización de las pruebas así como sus datos transcendentales para el trabajo.

- Capítulo 5 – Conclusiones y trabajos futuros
    - Por un lado se explican las conclusiones que se han podido conseguir gracias a los resultados previamente obtenidos y por otro lado los trabajos futuros. En cuanto a trabajos futuros se exponen unas líneas de trabajo en las que este algoritmo tendría proyección y sus puntos débiles para poder mejorarlos en un futuro.
- Capítulo 6 – Planificación y presupuesto
    - Este capítulo abarca la planificación del proyecto y el presupuesto económico de la ejecución del mismo.

Capítulo 2
Estado del arte

Capítulo 2

ESTADO DEL ARTE

En este apartado se expone el estado del arte relativo a la detección y seguimiento de personas en secuencias de vídeo.

Se empieza con una breve descripción de los conceptos básicos que hay que conocer a la hora de trabajar con imágenes. A continuación se detallan de forma breve y concisa algunas de las técnicas empleadas en el procesamiento de imágenes digitales. En último lugar se describen las diferentes metodologías para realizar el seguimiento de personas en secuencias de vídeo.

2.1. INTRODUCCIÓN A LA IMAGEN DIGITAL

La aparición del término píxel surgió a partir de un acrónimo de dos términos: *pix* (expresión coloquial que se refiere a *picture*) y *element*. Este término se utiliza en ambientes de la informática y fotografía digital para referirse a la menor unidad homogénea en color que forma una imagen [22].

Podemos decir que una imagen digital está formada por un conjunto de píxeles, cada uno con una intensidad y brillo distinto. Una imagen digital se representa a partir de una matriz numérica bidimensional en la que cada coeficiente de la matriz se corresponde con un píxel de la imagen.

Las imágenes digitales se pueden clasificar en función del rango de valores que pueden tomar los píxeles que la componen. Se tiene:

- Imágenes binarias: Imágenes que se obtienen a partir de la binarización de imágenes de niveles de gris. Se caracterizan porque sólo pueden estar formadas por dos valores: 0 para el negro y 1 para el blanco.
- Imágenes en escala de grises: En esta clase de imágenes cada píxel tiene un valor numérico, desde 0 (negro) hasta 255 (blanco) para representar su luminancia.
- Imágenes en color: Cada píxel está formado por un vector de tres componentes (que varían según el espacio del color en el que estemos trabajando).
- Imágenes multibanda: Cada píxel está formado por un vector de N componentes, con N mayor de 3. Un ejemplo de imágenes multibanda serían las imágenes híper-espectrales captadas por satélites artificiales).

2.1.1. Espacios de color

A continuación se detallan los diferentes espacios de color en el que se puede trabajar cuando realizamos el tratamiento de imágenes [14].

2.1.1.1. RGB

El modelo de color RGB (siglas de *red*, *green* y *blue*) es el más conocido y el más utilizado normalmente para representar el color en sistemas de vídeo, cámaras y monitores de ordenadores. Se trata de un modelo de color basado en síntesis aditiva de las componentes de luminancia relativas al rojo, verde y azul.

La representación del modelo RGB se realiza en un cubo unitario mostrado en la Figura 1, en el que los valores RGB están en cada vértice; el cian, magenta y amarillo en otros tres vértices; el negro en el origen de coordenadas y el blanco en el vértice opuesto. Un píxel en el espacio RGB está formada por tres canales o planos, uno para cada componente de color primario. El valor de cada plano se expresa con un valor comprendido entre 0 y 255, por lo que se tienen 256 valores posibles para cada color primario.

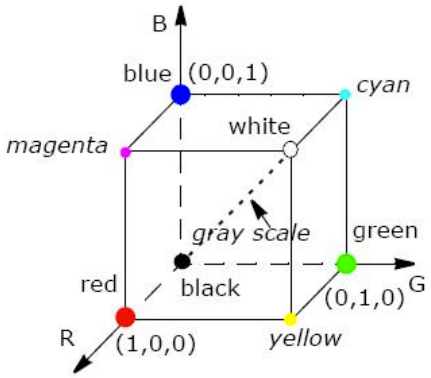


Figura 1 - Modelo de color RGB

2.1.1.2. HSV

El modelo HSV (del inglés *Hue, Saturation, Value*) o HSB (del inglés *Hue, Saturation, Brighness*) se basa en una transformación no lineal del espacio del RGB [6]:

$$H = \cos^{-1}\frac{\frac{1}{2}((R-G)+(R-B))}{\sqrt{((R-G)^2+(R-G)(R-B))}}$$

$$S = 1 - 3\frac{\min(R,G,B)}{R+G+B}$$

$$V = \frac{1}{3}(R+G+B)$$

Ecuación 1 – Transformación de RGB a HSV

El modelo HSV sigue un sistema de coordenadas cilíndrico, y el modelo subconjunto donde se define el color es una pirámide de base hexagonal.

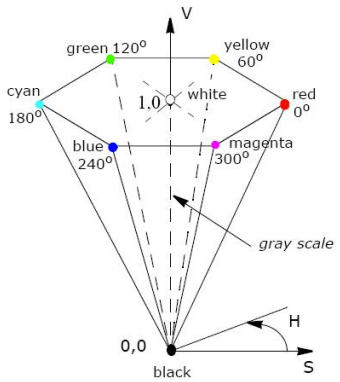


Figura 2 - Modelo de color HSV

El color o matiz, expresado con la letra H, se representa con un ángulo entre 0 y 360°. Cada valor representa un color, por ejemplo 0º representa el rojo, 60° el amarillo y 120° el verde.

La componente S representa la saturación, que es la cantidad de gris respecto al tono puro. Los valores posibles van desde el 0 (tono con mayor tonalidad grisácea) hasta el 1 (tono puro).

La componente V cuyo rango es del 0 al 1, indica la cantidad de luz que tiene un color.

2.1.1.3. YC_bC_r

YC_bC_r es una versión normalizada de los espacios de color para la televisión. El color está representado por su luminancia (Y) y por dos valores diferentes de color (C_b y C_r).

La luminancia, que puede ir desde el 0 hasta el 1, indica la luminosidad o claridad del color. El conjunto C_b-C_r sitúan el color en una escala: C_b entre el azul y el amarillo, y C_r entre el rojo y el verde. En la Figura 3 se puede apreciar el cubo RGB dentro del modelo de color YC_bC_r.

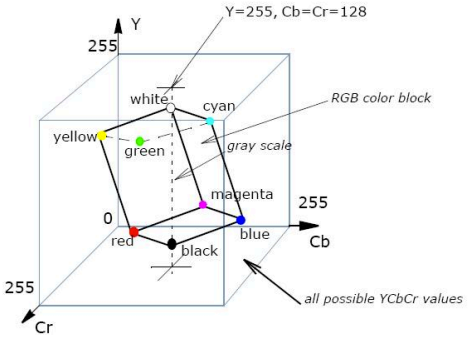


Figura 3 - RGB dentro del modelo YC_bC_r

Para obtener el equivalente de un color RGB en este modelo de color se pueden aplicar la siguiente ecuación [19]:

$$[Y \quad C_b \quad C_r] = [R \quad G \quad B]\begin{bmatrix}0.299 & -0.168 & 0.499\\ 0.587 & -0.331 & -0.418\\ 0.114 & 0.500 & -0.081\end{bmatrix}$$

Ecuación 2 – Transformación de RGB a YC_bC_r

La simplicidad de la transformación entre los diferentes modelos y la separación de la componente de luminancia de las de crominancia hacen que el espacio YC_bC_r un método atractivo para la modelización del color de la piel.

2.1.1.4. Otros modelos

- MODELO CMY

La mayoría de los dispositivos que depositan pigmentos coloreados sobre papel, como por ejemplo las impresoras, necesitan una entrada CMY (del inglés *Cyan, Magenta, Yellow*) o bien realizar la conversión interna de RGB a CMY. El sistema de coordenadas en el que se basa este modelo de color es el mismo que el de RGB con la peculiaridad que donde antes había negro ahora hay blanco y viceversa.

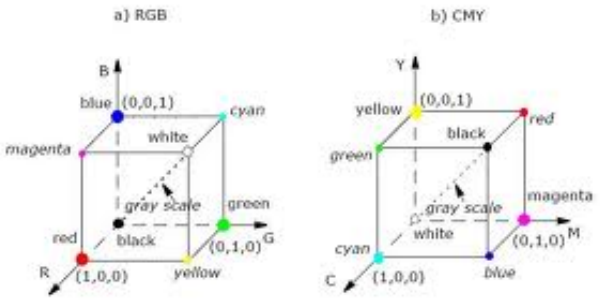


Figura 4 - a) Representación del modelo RGB b) Representación del modelo CMY

La conversión de RGB a CMY se realiza mediante esta simple ecuación:

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

Ecuación 3 – Transformación de RGB en CMY

- Modelo YUV

El modelo de color YUV es el modelo básico del color empleado en los sistemas de radiodifusión analógica de televisión en color (como PAL y NTSC). Principalmente YUV se creó para mejorar la eficiencia de la transmisión y para la compatibilidad hacía abajo con la televisión en blanco y negro. YUV define un espacio de color en función de una componente de luminancia (Y) y dos componentes de crominancia (U y V). La luminancia se puede calcular como una suma ponderada de las componentes RGB; las componentes de crominancia se forman restando la luminancia del azul y del rojo respectivamente.

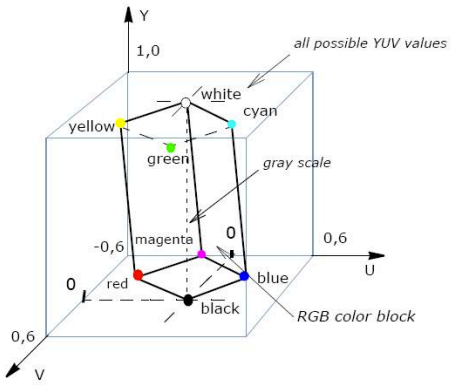


Figura 5 - RGB dentro del modelo YUV

2.2. TÉCNICAS DE PREPROCESADO

Las técnicas de preprocesado intentan optimizar o realzar las propiedades de las imágenes para una mejor interpretación en las siguientes etapas del análisis (en este proyecto la extracción de características). Estas técnicas se centran en la eliminación de ruido, realce de bordes, selección de los mejores valores de contraste-brillo y eliminación de los efectos de distorsión.

Estos métodos se pueden clasificar en dos grandes grupos en función de si trabajan con un único pixel o también tienen en cuenta el conjunto de píxeles que rodea al de estudio.

2.2.1. Operaciones puntuales

Son aquellas en las que el valor del píxel de la señal de la salida depende exclusivamente del valor del píxel de la señal de la entrada, por lo que no tienen en cuenta la vecindad con otros píxeles.

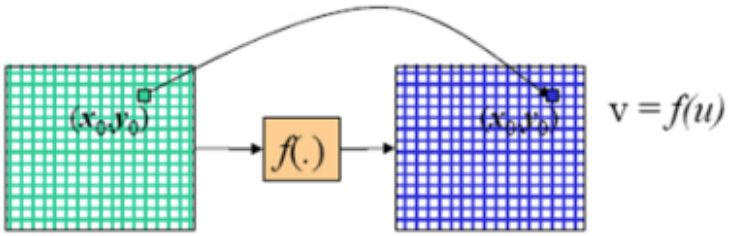


Figura 6 - Operaciones puntuales

Las operaciones puntuales más importantes son el contraste (y sus variedades como el recorte y la umbralización), la expansión o compresión del margen dinámico, la transformación de una imagen en su negativo, el modelado del histograma y las operaciones entre imágenes (suma, resta, producto, máximo…). Se estudiarán en más detalle el contraste (y sus variantes) y el modelado del histograma.

2.2.1.1. Contraste, recorte y umbralización

La operación de contraste comprime los niveles altos y bajos, reduciendo el rango dinámico. A su vez realza los niveles de gris medios otorgándoles mayor rango dinámico.

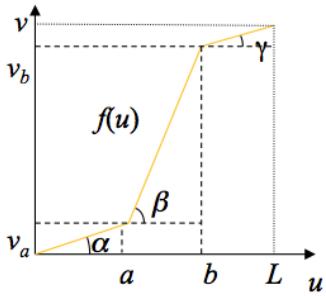


Figura 7 - Operación de contraste

$$v = \begin{cases} \tan(\alpha)\, u & 0 \le u < a \\ \tan(\beta)\,(u-a) + v_a & a \le u < b \\ \tan(\lambda)(u-b) & b \le u < L \end{cases}$$

Aplicando este procedimiento a una imagen, Figura 8, se puede observar que se expande la región del histograma donde se encuentran los valores de mayor aparición [10].

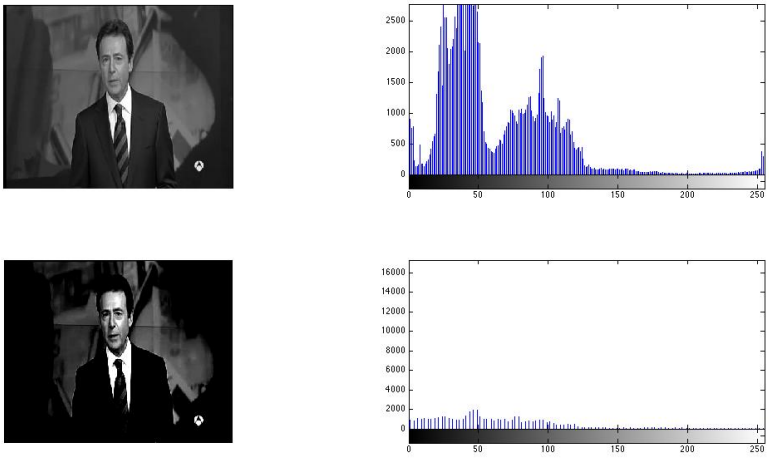


Figura 8 - Operación de contraste sobre fotograma

El recorte es un caso especial del aumento de contraste donde $\alpha = \lambda = 0$. En este caso los píxeles de entrada oscuros se convierten a negros, y los claros se transforman a blanco. Para los niveles intermedios se amplia el rango dinámico.

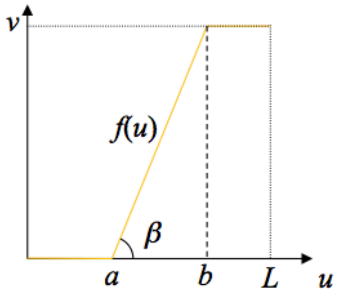


Figura 9 - Operación de recorte

Para la umbralización se tiene que $\alpha = \lambda = 0$ y $a = b = t$, donde t representa el umbral a partir del cual los niveles mayores pasan a convertirse en blanco y los niveles menores en negro.

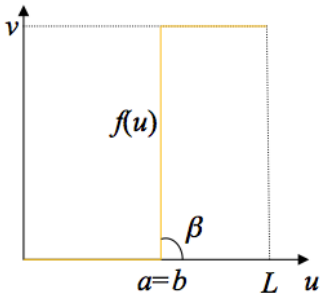


Figura 10 - Operación de umbralización

2.2.1.2. Modelado del histograma

Un histograma representa la frecuencia relativa de cada color que hay en una imagen. Si está normalizado tendrá valores entre 0 y 1, y se puede interpretar como la probabilidad de aparición de los niveles de gris. Hay dos operaciones significativas en este apartado: la especificación del histograma y la igualación del histograma.

Con la especificación del histograma lo que conseguimos es realzar ciertos niveles de gris especificando un histograma con una forma particular.

La igualación del histograma consiste en convertir cada nivel de gris de la imagen de entrada en otro de manera que el histograma de la imagen de salida sea totalmente uniforme. Generalmente se consigue mejorar la calidad de la imagen, aumentando el rango dinámico de los valores que aparecen mucho y reduciendo el de los valores que aparecen poco.

Esta operación, como cualquier otra de igualación de intensidad, se puede implementar mediante una tabla LUT (del inglés *Look-Up Table*) que consiste en una tabla que reasigna el nivel de gris de cada pixel.

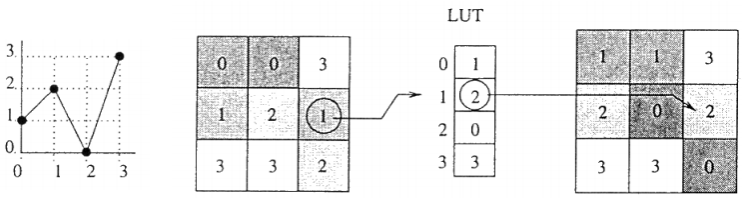


Figura 11 - Tabla LUT para igualación de histograma

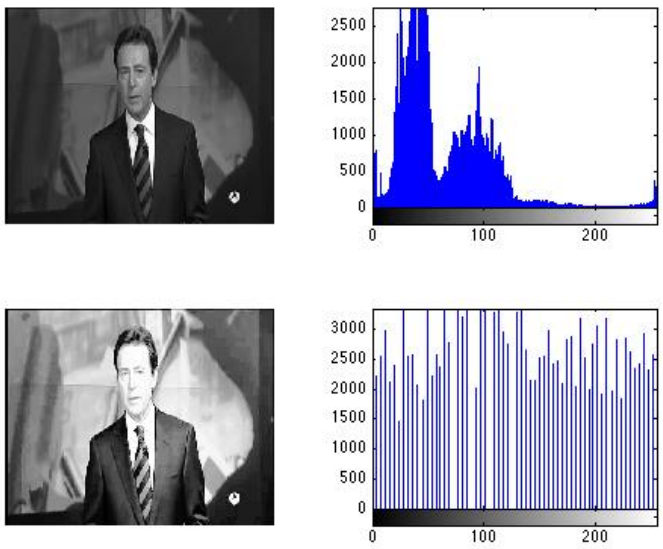


Figura 12 - Igualación del histograma para un fotograma

2.2.2. Operaciones espaciales

Las operaciones espaciales o de vecindad son aquellas operaciones en las que se modifica el valor del pixel de acuerdo a los valores de los pixeles que le rodean. Suelen basarse en máscaras espaciales (matriz de números) cuyos coeficientes determinan la transformación.

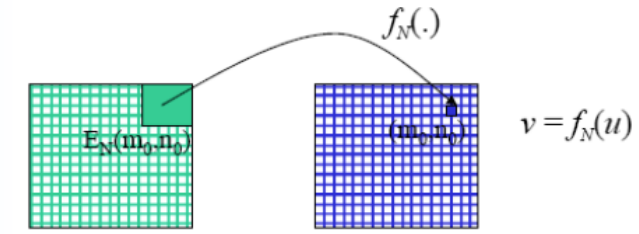


Figura 13 - Operaciones espaciales

2.2.2.1. Filtrado espacial

El filtrado espacial es la operación que se emplea en el procesamiento de las imágenes para mejorar o suprimir detalles espaciales como pueden ser los bordes o los patrones de ruido. Los filtros empleados en esta operación se pueden dividir en dos grupos, por un lado están los filtros lineales e invariantes en el tiempo (filtro paso bajo, filtro paso alto) y por otro lado los filtros no lineales (filtro máximo, filtro mínimo y filtro de mediana).

2.2.2.1.1. Filtro paso bajo

Los filtros paso bajo acentúan las bajas frecuencias, desenfocando la imagen y eliminado el ruido. Esto produce un difuminado (mayor cuanto más grande es la máscara del filtro) perdiéndose nitidez pero ganándose homogeneidad.

La máscara de clase de filtros se caracteriza porque todos los coeficientes son positivos y la suma tiene que dar como resultado la unidad.

$$\frac{1}{9}\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix} \qquad \begin{bmatrix}0 & 1/8 & 0\\ 1/8 & 1/2 & 1/8\\ 0 & 1/8 & 0\end{bmatrix}$$

Figura 14 - Máscaras 3x3 de un filtro paso bajo

2.2.2.1.2. Filtro paso alto

Los filtros paso alto realizan la operación contraria a los filtros paso bajo, por lo que enfatizan las altas frecuencias resaltando los bordes y eliminan las bajas frecuencias. A su vez la imagen se ve mejor enfocada.

La suma de los coeficientes de la máscara de este clase de filtros debe dar como resultado 0. Algunos ejemplos pueden ser los siguientes:

Figura 15 - Máscaras 3x3 de un filtro paso alto

Los filtros detectores de bordes son lo que principalmente se encargan de realzar los contornos en una imagen. Normalmente estos filtros crean una imagen con fondo gris y líneas blancas y negras rodeando los bordes de los objetos. Los operadores más utilizados son: Roberts, Prewitt y Sobel [4], [30].

- Operador de Roberts. Las máscaras utilizadas en este caso son:

Figura 16 - Operador Roberts para un gradiente de fila y para un gradiente de columna

Con este operador se obtienen buenos resultados a la hora de detectar bordes diagonales. El gran inconveniente que presenta es su extremada sensibilidad al ruido.

- Operador de Prewitt: Para este caso se utiliza la máscara que se muestra a continuación con K=1:

Figura 17 - Operador Prewitt para un gradiente de fila y para un gradiente de columna

Este operador proporciona una mejor detección de los bordes verticales y horizontales en comparación con los bordes diagonales.

- Operador de Sobel. Se utiliza de nuevo la máscara de la Figura 17 pero ahora con K=2. Este operador también tiene una buena respuesta ante los bordes verticales y horizontales. Además realiza un suavizado en la imagen.

2.2.2.1.3. Filtro de mediana

Se utiliza para eliminar el ruido manteniendo los bordes de la imagen. La principal ventaja que presenta es que da muy buenos resultados cuando se trata de ruido con características impulsivas. Hay que tener en cuenta que no es un filtro lineal e invariante.

2.2.2.2. Filtros morfológicos

Se basan en una serie de operaciones morfológicas básicas que actúan tomando las características geométricas y topológicas del elemento estructurante, es decir, se trabaja en el dominio espacial.

El elemento estructurante, dual a una máscara de convolución, es un patrón de ajuste que se usa para examinar la estructura de la imagen. La forma y el tamaño del elemento estructurante permite determinar la forma de los objetos de la imagen.

Las transformaciones morfológicas más interesantes son [9]:

- Dilatación: Su objetivo es rellenar pequeños agujeros de tamaño igual o menor que el elemento estructurante.

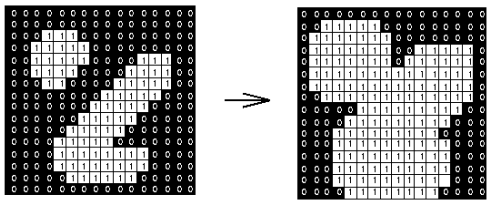


Figura 18 - Efecto de la dilatación con un elemento estructurante 3x3

- Erosión: Elimina los grupos de píxeles donde el elemento estructurante no entra, como islas pequeñas o protuberancias.

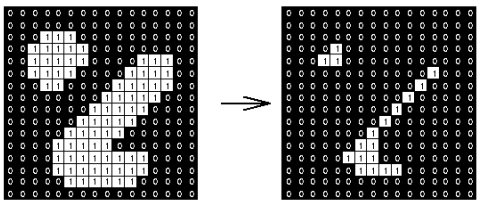


Figura 19 - Efecto de la erosión con un elemento estructurante 3x3

- Apertura y cierre: La apertura es una combinación de erosión más dilatación, alisa contornos, elimina protuberancias y suaviza los bordes. El cierre es una combinación de dilatación más erosión, rellena vacíos, elimina entrantes y conecta objetos vecinos.

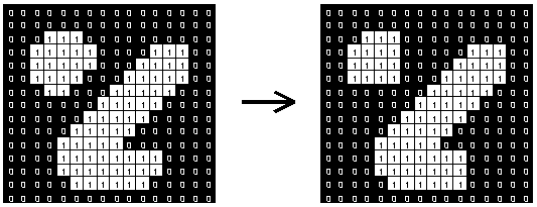


Figura 20 - Efecto de la apertura con un elemento estructurante 3x3

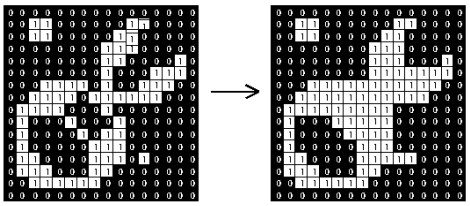


Figura 21 - Efecto del cierre con un elemento estructurante 3x3

2.3. TRANSFORMADAS DE LA IMAGEN

Las operaciones de las que se ha estado hablando hasta ahora se realizan en el dominio espacial, es decir, trabajando con los niveles de intensidad de los píxeles y sus relaciones posicionales. Existe también la posibilidad de trabajar en el dominio de la frecuencia [3].

La teoría de las transformadas adquiere un papel clave en el procesamiento de las imágenes durante muchos años. A día de hoy sigue siendo un tema de especial interés tanto en la parte teoría como en la parte aplicativa.

Aunque existen otras transformadas en este capítulo trata principalmente del desarrollo de la transformada de Fourier discreta bidimensional y de sus propiedades pues es la transformada con mayor variedad de aplicaciones en

problemas de procesamiento de imágenes [12]. Esta transformada es una representación de la imagen como suma de exponenciales complejas de distintas amplitudes, frecuencias y fases, que definen los cambios espaciales de la imagen.

2.3.1. La transformada de Fourier discreta bidimensional

Para poder utilizar la transformada de Fourier sobre una imagen digital en lugar de una función continua hay que considerar la relación que existe entre estos dos términos: una imagen digital se obtiene a partir del muestreo de una señal. Vamos a estudiar el efecto que el proceso de muestreo tiene sobre el cálculo de la transformada de Fourier.

En el caso de dos variables, el par de la transformada discreta de Fourier vendrá definidas por las siguientes expresiones:

$$F(u,v) = \frac{1}{NM}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)\exp\left[-2j\pi\left(\frac{ux}{M}+\frac{vy}{N}\right)\right]$$

Ecuación 4 – Transformada de Fourier

para $u = 0,1,\dots,M-1\ y\ v = 0,1,\dots,N-1, y$

$$F(x,y) = \sum_{x=0}^{M-1}\sum_{y=0}^{N-1} F(u,v)\exp\left[-2j\pi\left(\frac{ux}{M}+\frac{vy}{N}\right)\right]$$

Ecuación 5 – Transformación de Fourier inversa

para $u = 0,1,\dots,M-1\ y\ v = 0,1,\dots,N-1, y$

Como nos encontramos en el caso bidimensional los incrementos de muestreo en los dominios espacial y de frecuencia están relacionados por:

$$\Delta u = \frac{1}{M\Delta x}\ y\ \Delta v = \frac{1}{N\Delta y}$$

Ecuación 6 – Incrementos de muestreo

Como en la práctica es normal que las imágenes se digitalicen en matrices cuadradas, es decir M=N, las dos ecuaciones descritas anteriormente se transformarían en:

$$F(u,v) = \frac{1}{N}\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y)\exp\left[-2j\pi\left(\frac{ux}{N}+\frac{vy}{N}\right)\right]$$

Ecuación 7 – Transformada de Fourier para una distribución cuadrada

para $u, v = 0,1, \ldots, N-1$ y

$$F(x,y) = \frac{1}{N}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} F(u,v)\exp\left[-2j\pi\left(\frac{ux}{N}+\frac{vy}{N}\right)\right]$$

Ecuación 8 – Transformada de Fourier inversa para una distribución cuadrada

El resultado de aplicar la transformada de Fourier sobre una imagen Figura 22 debe ser otra imagen de igual dimensión, pero el rango de valores de los píxeles cambia considerablemente porque el rango dinámico de la transformada de Fourier es mayor al rango típico de las imágenes.

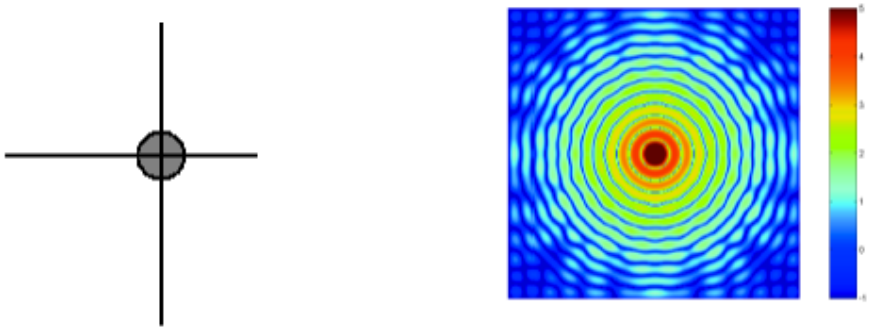


Figura 22 - Izquierda: imagen original. Derecha: transformada de Fourier

Algunas de las propiedades que presenta la transformada de Fourier bidimensional son: la separabilidad, la traslación, la periodicidad, la simetría conjugada, la rotación, la distributiva y el cambio de escala. Dos operaciones espaciales muy interesantes a la hora de trabajar con la transformada de Fourier en el campo del procesamiento de imágenes son la convolución y la correlación, nos centraremos en la correlación que es la que se emplea en la implementación de este trabajo.

2.3.1.1. Correlación

La correlación [12] o correlación cruzada, si f(x) y g(x) son distintas, de dos funciones continuas unidimensionales f(x) y g(x) se define por la relación:

$$f(x) \circ g(x) = \int_{-\infty}^{\infty} f^*(\alpha) g(x+\alpha) d\alpha$$

Ecuación 9 – Correlación entre dos funciones continuas

donde * es el conjugado complejo.

La expresión de la correlación en funciones discretas se define como:

$$f(x) \circ g(x) = \sum_{m=0}^{M-1} f^*(M) g(x+M)$$

Ecuación 10 – Correlación entre dos funciones discretas

para $x = 0,1,\dots,M-1$.

Para el caso bidimensional siguen siendo válidas expresiones similares:

$$f(x,y) \circ g(x,y) = \int \int_{-\infty}^{\infty} f^*(\alpha,\beta) g(x+\alpha, y+\beta) d\alpha d\beta$$

Ecuación 11 – Correlación bidimensional entre dos funciones continuas

$$f(x,y) \circ g(x,y) = \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} f^*(m,n) g(x+m, y+n)$$

Ecuación 12 – Correlación bidimensional entre dos funciones discretas

para $x = 0,1,\dots,M-1$ *e* $y = 0,1 \dots N-1$

Una de las más importantes aplicaciones de la correlación en el ámbito del procesado de imágenes es hallar réplicas de una subimagen dentro de una imagen, donde el problema consiste en hallar el mayor parecido entre las dos imágenes.

De una forma simple, la correlación entre la imagen f(x,y) de dimensiones MxN y la subimagen w(x,y) de tamaño JxK se puede expresar como:

$$c(s,t) = \sum_x \sum_y f(x,y) w(x-s, y-t)$$

Ecuación 13 – Correlación entre imagen f(x,y) y subimagen w(x,y)

donde s=0,1,2,…,M-1, t=0,1,2,…,N-1 y el sumatorio se calcula para el área de solapamiento.

La Figura 23 muestra el procedimiento de la correlación. Para cualquier valor de (s,t) dentro de f(x,y) la Ecuación 13 dará un valor de c. Al variar s y t, w(x,y) se desplazará por la imagen, dado lugar al área de solapamiento, y obteniéndose una función c(s,t). El máximo valor de c(s,t) determinará la posición en la que se produce la mayor correspondiente entre la imagen y la sub-imagen.

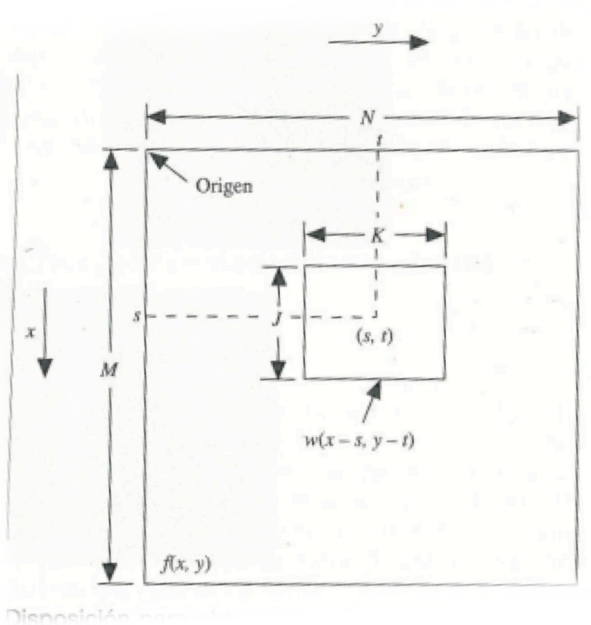


Figura 23 - Procedimiento de correlación

2.4. SEGMENTACIÓN

La segmentación de imágenes consiste en dividir la imagen en regiones en las que los elementos constituyentes, los píxeles, tengan características similares. Cada una de estas regiones de interés se denomina objeto, por lo que aplicando la operación de segmentación a una imagen se obtiene una separación de objetos [16].

2.4.1. Segmentación por fronteras

Los métodos de segmentación por fronteras aplican a la imagen un detector de bordes. Las regiones se definen a partir de las fronteras delimitadas por los bordes detectados.

A la hora de detectar los bordes se pueden seguir dos estrategias: la detección de máximos en derivadas primeras o la detección de cruces por derivadas segundas. Los detectores de bordes basados en derivadas primeras aproximan el cálculo del módulo del gradiente mediante operadores discretos, y asignan al borde a aquellos píxeles cuyo gradiente es superior a un umbral. Cuando los bordes no están muy marcados se pueden emplear las derivadas de segundo orden porque los bordes en una imagen continua son puntos de inflexión.

2.4.2. Segmentación por regiones

A cada píxel se le asigna una región en función de las características locales de la imágenes en el píxel en estudio y las características de los píxeles vecinos. Se destacan tres grandes tipos: segmentación por umbral, segmentación por agrupamiento y segmentación por evolución de regiones.

2.4.2.1. Segmentación por umbral

La umbralización [12] es una de las técnicas más importantes de la segmentación de imágenes. La umbralización se puede considerar como una operación que implica realizar comprobación frente a una función T de la forma:

$$T = T[x, y, p(x, y), f(x, y)]$$

Ecuación 14 – Función de umbralización

donde f(x,y) es el nivel de gris del punto (x,y) y p(x,y) representa alguna propiedad local de este punto. La imagen umbralizada resultante g(x,y) se obtiene a partir de:

$$g(x, y) = \begin{cases} 1 \; si \; f(x, y) > T \\ 0 \; si \; f(x, y) \leq T \end{cases}$$

Ecuación 15 – Imagen umbralizada g(x,y)

La técnica más sencilla de todas las técnicas de umbralización es la umbralización por histograma utilizando un umbral T. La operación consiste en etiquetar cada píxel de la imagen en función de si es perteneciente o no al objeto del fondo, dependiendo de que el nivel de gris de ese pixel sea mayor o menor que el valor de T.

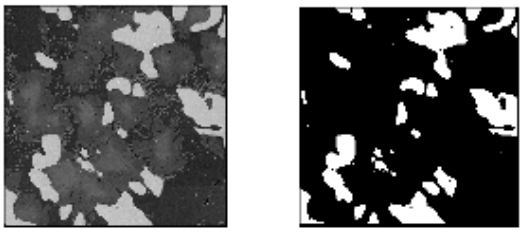


Figura 24 - Segmentación por umbral

2.4.2.2. Segmentación por agrupamiento

La segmentación por agrupamiento [16] permite seleccionar varias características a la vez, lo que da como resulta la posibilidad de segmentar la imagen en más de dos regiones.

Tiene dos principios básicos en los que se fundamenta:

- Cada región del espacio de características se define mediante un patrón o centroide.
- Cada vector de características se asigna a la región del centroide más próximo.

Esta versión de segmentación es útil cuando no se conocen las características de las regiones que se buscan ni tampoco cuantas regiones hay.

El agrupamiento o *clustering* se puede realizar a través del algoritmo de k-medias. Se inicia el método con k muestras aleatorias, se calcula la distancia de cada dato a cada uno de los k representantes y se asigna a aquel que guarde mínima distancia. Se recalculan los representantes de cada clase y se vuelve a aplicar el algoritmo de forma iterativa.

Cuando a priori no se conoce el número de regiones de la imagen, se define una partición inicial por ejemplo de dos centroides (y por lo tanto dos regiones), a continuación se aplica el algoritmo k-medias. Si las regiones son lo suficientemente homogéneas en base al criterio (condición de parada) que se hayan definido se detiene el algoritmo; en caso contrario se divide la región menos homogénea y se vuelve a aplicar el algoritmo de forma iterativa.

2.4.2.3. Segmentación por evolución de regiones

Los métodos por evolución de regiones parten de una segmentación inicial que se va modificando con el objetivo de obtener regiones con las propiedades deseadas.

El crecimiento de regiones es una metodología que agrupa píxeles o subregiones dentro de regiones más grandes de acuerdo con un criterio predefinido. Se empieza con un conjunto de puntos a partir de los que van creciendo las regiones al agregar a éstas píxeles vecinos con propiedades similares (como el nivel de gris, textura, color).

Este método presenta dos problemas. Al agrupar píxeles con la mismas propiedades para formar una región sin tener en cuenta la conectividad puede generar una segmentación que no tenga sentido en el contexto en el que se está trabajando. Otro problema del crecimiento de regiones es la formulación de una regla de parada; el crecimiento de una región debe detenerse cuando no hay más píxeles que satisfagan el criterio de inclusión en esa región.

Figura 25 - Segmentación basada en evolución de regiones

2.4.3. Segmentación basada en modelos

Como se ha comentado, las técnicas anteriores emplean en mayor medida información local. La información global se introduce a través de modelos. La segmentación basada en modelos [20] consiste en enlazar los bordes locales cuando forman parte de una curva específica como una recta, un rectángulo, una circunferencia… Para la localización se utilizan técnicas basadas en la Transformada de Hough que además tiene buenos resultados en la segmentación de objetos solapados o parcialmente ocluidos.

2.5. TRACKING

El *tracking* se conoce como la acción de seguir al objeto en estudio en todos los *frames* que componen un vídeo. Debido al amplio rango que esta aplicación posee se han desarrollado numerosas investigaciones relacionas con el tema del tracking de personas.

Los algoritmos de seguimiento que se han desarrollado dependen en su mayoría de la aplicación en la que se vayan a aprovechar. En el apartado siguiente se hablará de los algoritmos más relevantes.

2.5.1. Algoritmos de tracking

Revisando proyectos anteriores [15], [31] se observa que atendiendo al procesamiento de las imágenes los algoritmos de *tracking* se pueden clasificar en base al uso o no de modelos de forma para seguir al individuo en estudio. También existe una tercera categoría que engloba los algoritmos que se basan en la utilización de filtros de predicción o estimación.

$$Algoritmos\ de\ tracking \begin{cases} Basados\ en\ modelos \\ No\ basados\ en\ modelos \\ Basados\ en\ predicción\ de\ filtros \end{cases}$$

Figura 26 - Algoritmos de tracking

La primera clase, algoritmos de *tracking* basados en modelos, se refiere a aquellas técnicas donde el seguimiento de un individuo se basa en la comparación de cada imagen con un patrón. Este tipo de técnicas se basan en el conocimiento de un modelo predeterminado del individuo (que se construyen off-line con mediciones manuales), se extraen las características y se asocian con la estructura del modelo y del movimiento. Como se puede notar esta tarea tiene un coste computacional muy elevado y requiere una fuerte segmentación el individuo desde el *background*, por lo que se considera una técnica difícil de implementar.

A raíz de los problemas descritos anteriormente surgió el *tracking* no basado en modelos. Este tipo de tracking se basa en las características extraídas de la imagen y no en la búsqueda de un patrón. Este planteamiento suele ser más eficiente computacionalmente hablando que el anterior pero presenta menos robustez.

Por último nos encontramos con los algoritmos de tracking basados en predicciones y medidas destacando entre ellos el filtro de Kalman [24], el algoritmo de condensación [26], el *mean shift* y las técnicas de filtrado bayesiano [27].

A continuación vamos a hablar de los algoritmos de tracking no basados en modelos, base de este trabajo.

2.5.1.1. Algoritmos de tracking no basados en modelos

Como ya se ha comentado los algoritmos basado en modelos son bastante eficaces y robustos, sin embargo su coste computacional es demasiado elevado. Se desarrollaron entonces los algoritmos de tracking no basados en modelos para los que es innecesario conocer de antemano la información estructural del objeto a seguir.

Los principales algoritmos de esta clase son:

- Seguimiento basado en regiones.
- Seguimiento basado en contornos activos.
- Seguimiento basado en rasgos.

➢ SEGUIMIENTO BASADO EN REGIONES

Este tipo de algoritmos identifican un *blob* o región conectada con el espacio que se asocia a cada objeto en estudio y se sigue utilizando una medida de similitud o un parámetro de correlación[1]. El fondo de la imagen tiene que ser calculado y mantenido dinámicamente y las regiones de movimiento se detectan por sustracción de este.

Un ejemplo del uso de un algoritmo de esta clase se puede ver en [8]. Utiliza la correspondencia de matrices, *matching*, en dos direcciones empleando el criterio de que cuando dos *bounding boxes*, o cajas de correspondencia, se solapan en un mismo frame se hace un *matching* del *blob* resultante en los siguientes frames.

➢ Seguimiento basado en contornos activos.

Los algoritmos basados en contornos activos, también llamados *Snake*, realizan el tracking de los objetos representando sus contornos como bordes bien delimitados y actualizándolos al pasar de un frame a otro.

En [1] sus autores describen un modelo variacional que minimiza la energía para ajustar una curva C deformable a los contornos de una imagen.

$$\mathbf{E_{KWT}}(\mathbf{C}) = \alpha \int_0^1 |\mathbf{C}'(\tau)|^2 \, \mathbf{d}\tau + \beta \int_0^1 |\mathbf{C}''(\tau)|^2 \, \mathbf{d}\tau - \lambda \int_0^1 |\nabla \mathbf{I}(\mathbf{C}(\tau))| \, \mathbf{d}\tau$$

Ecuación 16 – Energía del Snake

Los dos primeros términos de la función descrita anteriormente determinan la regularidad de las fronteras que se van a detectar y reciben el nombre de energía interna. El tercer término representa la energía de la imagen y recibe el nombre de energía externa. Resolver este algoritmo significa minimizar E_{KWT}, se utiliza normalmente el método del descenso de energía, partiendo de una situación inicial movemos la curva siguiendo la dirección de máximo decrecimiento de la energía.

Aunque este método es computacionalmente menos costoso que el seguimiento basado en regiones presenta varias limitaciones porque tiene dificultades a

[1] Según se explica más adelante, la técnica usada en este trabajo.

la hora de delimitar el contorno inicial y no puede haber cambios en la topología.

- Seguimiento basado en rasgo.

En lugar de hacer el seguimiento basándose en una región o en los contornos, este método utiliza los rasgos sobresalientes de los elementos característicos y luego hace una correspondencia entre las imágenes [15]. Existen dos amplias aproximaciones para este clase de algoritmos: seguimiento de rasgos dinámicos y seguimiento de rasgos estáticos.

Cuando se emplea el método estático, los rasgos se extraen a priori independientemente en cada frame y el algoritmo calcula la correspondencia óptima entre ellos. En el seguimiento de rasgos dinámicos los rasgos son determinados y seguidos de forma dinámica sobre la secuencia de frames.

Capítulo 3
Diseño e implementación

3.1. ARQUITECTURA DEL SISTEMA

El objetivo principal de este proyecto es el desarrollo y la implementación de un sistema automático capaz de detectar personas en secuencias de video. En general, la mayoría de los sistemas de reconocimiento estudiados [9], [13] tienen la misma secuencia de etapas, mostrada en la figura siguiente.

Figura 27 - Diagrama de bloques de un sistema de reconocimiento

Las características que diferencian el algoritmo desarrollado en este trabajo de resto son:

- Se parte de una secuencia de vídeo en lugar de una serie de imágenes independientes unas de otras.

- No existe un bloque de extracción de características como tal y el etiquetado consiste únicamente en decir si la persona a detectar se encuentra o no en el frame en estudio.

A continuación se muestra un diagrama en el que es pueden reconocer visualmente los bloques que forman el sistema seguido de un breve resumen de lo que implica da uno de ellos. En cada uno de los apartados de este capítulo se desarrollará cada bloque, explicando tanto su funcionalidad como las funciones que lo componen.

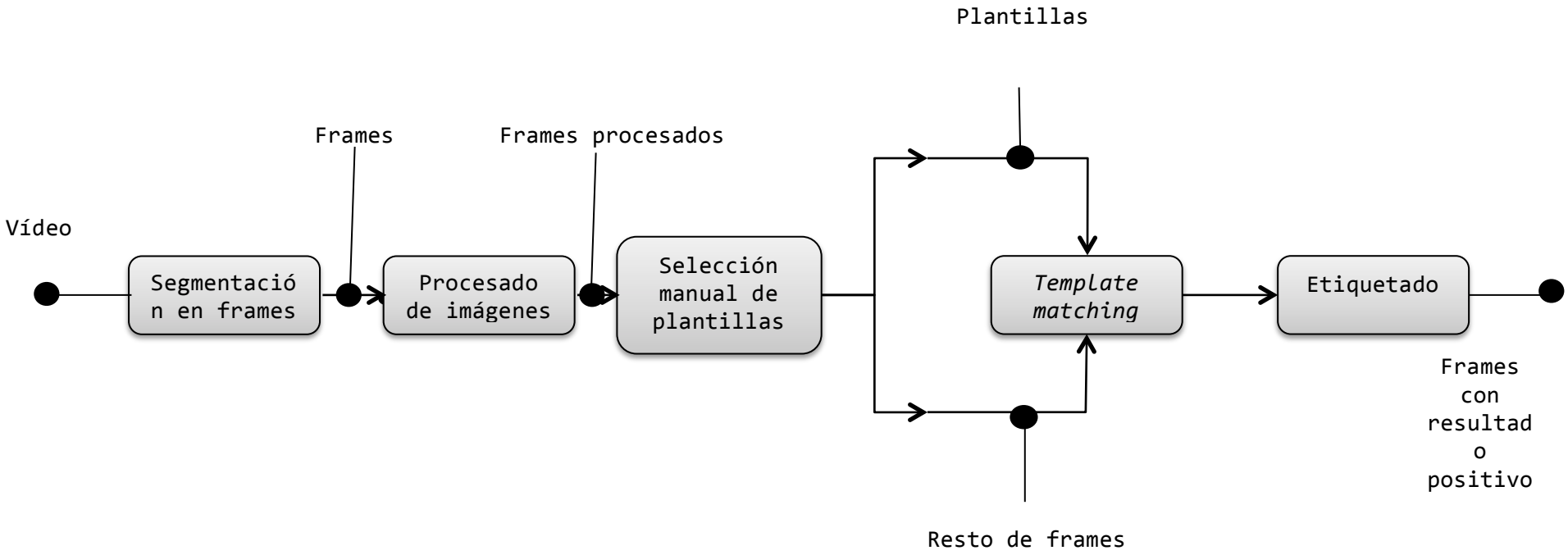


Figura 28 - Diagrama de bloques del algoritmo desarrollado

- Segmentación en frames
  - A partir del vídeo de entrada se extraen los frames que lo componen para posteriormente poder detectar en cada uno de ellos a la persona de interés.

- Procesado de imagen
  - El parámetro de entrada de este bloque es cada uno de los frames que han sido extraídos del vídeo. En esta sección se aplican algunas de las técnicas explicadas en el capítulo 2 del estado del arte que facilitan la detección de la persona de interés.

- Extracción de las plantillas
  - De todos los frames que se han obtenido el usuario etiqueta los que se deseen como plantillas con las que realizar el *template matching*.

- *Template matching*
  - Una vez que se tienen los frames del vídeo procesados estos pasa al siguiente bloque que es donde, a partir de unas técnicas de *template matching* se realiza la detección de la persona entre las plantillas etiquetadas y el resto de frames.

- Etiquetado
  - Únicamente se etiqueta cada una de las imágenes en función de si contiene o no a la persona que se quiere detectar.

3.2. HERRAMIENTA UTILIZADA

Se ha utilizado la herramienta de software Matlab para el desarrollo de este proyecto.

Matlab (abreviatura de *MATrix LABoratory*) [28] es un entorno de cálculo matemático con un lenguaje de programación propio, lenguaje M. Integra: cálculo matricial, representación de datos y funciones, implementación de algoritmos, procesamiento de señales y creación de interfaces de usuario. Además permite comunicarse con otros programas, en otros lenguajes y con diferentes dispositivos hardware.

Matlab contiene dos herramientas adicionales que complementan sus prestaciones: *Simulink*, una plataforma de simulación multidominio, y GUIDE, un editor de interfaces de usuario. Si se quieren ampliar las capacidades de Matlab se pueden usar las diferentes cajas de herramientas, *toolboxes,* entre las que cabe destacar la referente al procesamiento de imágenes *Image Processing Toolbox*.

3.3. EXTRACCIÓN DE LOS FRAMES

Como ya se ha comentado, al partir de una secuencia la primera operación que se debe realizar es extraer los frames que contiene el vídeo con el que se va a trabajar.

Este proceso se lleva a cabo en la función getFrames. Ésta recibe un como parámetro de entrada un vídeo, del que obtiene sus características principales como la duración, el número de frames que componen el vídeo y el ratio de *frames per second* (FPS). Este último es el que se utiliza para la extracción de un frame cada segundo.

Para la implementación de esta etapa se han usado las funciones de la tabla siguiente:

Nombre de la función	**Parámetros de entrada**	**Descripción**	**Parámetros de salida**
OBJ=VideoReader(FILENAME)	FILENAME: nombre del archivo del vídeo que se quiere leer.	Construye un objeto a partir de los datos del vídeo.	OBJ: objeto.
value=get(OBJ, Name);	OBJ: objeto del que se quiere extraer las propiedades. Name: Si toma el valor 'NumberOfFrames' devuelve el número de frames. En cambio, 'FrameRate' devuelve el número de frames por segundo.	Método que devuelve el valor de la propiedad que especifica Name para el objeto OBJ.	Value: valor de la propiedad.
video=read(OBJ,INDEX)	OBJ: objeto con el que se quiere trabajar. INDEX: especifica los frame(s) con los que se quiere trabajar	Lee el(los) frame(s) enumerado(s) en INDEX del vídeo OBJ.	Video: matriz del objeto leído.
imwrite(A,FILENAME,FMT)	A: matriz que equivale a la imagen. FILENAME: nombre con el que se quiere guardar la imagen. FMT: formato con el que se quiere guardar	Escribe la imagen especificada por A, con el nombre indicado por FILENAME y el formato especificado por FMT.	

la imagen.		

Tabla 1 - Implementación getFrames

3.4. PROCESADO DE LAS IMÁGENES

Como se dijo en el capítulo 2 del estado del arte, las técnicas de procesado se emplean para mejorar las características de las imágenes y así facilitar la etapa de detección.

Al trabajar con vídeos de no muy alta calidad, la cantidad de pixeles de la imagen que tienen un valor que no se asemeja al valor de sus píxeles vecinos es muy elevada por lo que se tuvieron que emplear técnicas de eliminación de ruido aleatorio. Para reducir al máximo posible la componente impulsiva del ruido existen dos técnicas: filtrado paso bajo (en 2.3.2.1.1) o filtro de mediana (en 2.3.2.1.3).

El inconveniente que presentan los filtros paso bajo al realizar la eliminación del ruido es que los bordes se vuelven borrosos, y esta parte de la imagen siempre es muy útil pues contiene mucha información. En cambio en el filtro de mediana, al reemplazar el nivel de intensidad del pixel por la mediana de los niveles de intensidad de los pixeles vecinos los valores extremos no influyen de igual manera que al aplicar la media. En el ejemplo numérico que se muestra a continuación se puede ver la diferencia entre los dos métodos:

$$\begin{pmatrix} 20 & 83 & 57 \\ 62 & 71 & 90 \\ 86 & 76 & 80 \end{pmatrix} \rightarrow (20, 57, 62, 71, 76, 80, 83, 86, 90) \rightarrow \begin{cases} media = 69 \\ mediana = 76 \end{cases}$$

Para acentuar aún más los bordes de las imágenes y aumentar la nitidez de la misma se utilizó un filtro paso alto. En este caso se eligió usar el método de *Canny* pues al utilizar dos umbrales es capaz de detectar tanto bordes fuertes como débiles, e incluye los bordes débiles en la imagen de salida si están conectados con otros bordes fuertes. Por lo tanto este método es menos propenso que los demás a verse influido por el ruido.

La tabla que se muestra a continuación contiene las funciones que se utilizaron para el desarrollo de esta sección.

Nombre de la función	Parámetros de entrada	Descripción	Parámetros de salida
B=medfilt2(A,[M N])	A: matriz sobre la que se quiere realizar el filtrado. [M N]: indican el tamaño de la máscara con la que se realiza el barrido.	Realiza un filtrado de mediana.	B: matriz de salida.

BW=edge(I,'canny')	I: matriz sobre la que se realiza la operación. 'canny': especifica el método Canny.	Detección de bordes usando el método Canny.	BW: matriz de salida.

Tabla 2 - Implementación procImg

3.5. OBTENCIÓN DE LAS PLANTILLAS

Independientemente del número plantillas con las que se vaya a trabajar todas se extraen de la misma forma, manualmente. Es el usuario del sistema quién marca que patrones se quieren usar para realizar la detección de la persona en los distintos frames.

El usuario realiza una observación inicial de los frames que se han obtenido en la sección anterior y escoge aquel que cumpla los siguientes tres requisitos:

- Si el usuario quiere seleccionar un patrón para la cara, en el frame seleccionado se tiene que ver toda la cara de la persona, o la mayoría. Es decir, evitar frames en los que no se distinga la cara o muestren perfiles.

- Si el usuario quiere seleccionar un patrón para la vestimenta, o cualquier otra característica que diferencie a la persona en estudio del resto, como por ejemplo una gorra, el área que seleccione debe ser lo más preciso posible omitiéndose en la medida de lo posible detalles de fondo.

- La imagen seleccionada para extraer de ella la o las plantillas debe ser lo más nítida posible, estando la parte de la persona o característica a detectar lo menos influenciada por luces y/o sombras.

Para esta operación se utiliza la función de Matlab que se describe en la Tabla 3.

Nombre de la función	Parámetros de entrada	Descripción	Parámetros de salida
I2=imcrop(I, rect)	I: matriz sobre la que queremos realizar el recorte. rect: vector de cuatro elementos [xmin ymin width height] que especifica el tamaño y la posición del rectángulo de recorte.	Recorta la imagen I según lo especificado en rect.	I2: matriz recortada

Tabla 3 – Función imcrop

3.6. TEMPLATE MATCHING

La idea básica del *template matching* consiste en encontrar las ocurrencias de una plantilla dentro de una imagen más grande. Esto es, se mueve la plantilla o patrón sobre la imagen y se encuentra la mejor posición donde coincidan, utilizando la correlación. Los valores altos de correlación indican una buena correspondencia entre la imagen y el patrón, lo que se puede llegar a entender como una medida de similitud.

Los problemas que presenta la correlación son:

- Sensibilidad al ruido, unos pocos valores erróneos pueden cambiar la correlación significativamente.
- La correlación no es invariante a cambios de intensidad, tal como las condiciones de iluminación.
- Es variante a rotaciones y cambios de escala. Lo que quiere decir que dentro de la imagen en estudio la plantilla debe aparecer con la misma orientación y el mismo tamaño, porque sino no es capaz de detectarlo.
- Es computacionalmente caro si se usan plantillas grandes.

Se puede normalizar la correlación para minimizar los efectos del cambio de intensidad, se conoce como correlación normalizada. En Matlab este procedimiento se puede encontrar en la función normxcorr2.

Normxcorr2 utiliza el procedimiento siguiente:

- Calcula la correlación cruzada en el espacio o en el dominio de la frecuencia dependiendo del tamaño de las imágenes con las que trabaje.
- Calcula sumas locales calculando sumas continuas.
- Utiliza sumas locales para normalizar la correlación cruzada y obtener los coeficientes de correlación.

La implementación sigue de cerca la siguiente fórmula:

$$\boldsymbol{c} = \frac{\sum_{x,y}\left[\boldsymbol{f}(\boldsymbol{x},\boldsymbol{y}) - \bar{\boldsymbol{f}}\right][\boldsymbol{t}(\boldsymbol{x},\boldsymbol{y}) - \bar{\boldsymbol{t}}]}{\left(\sum_{x,y}[\boldsymbol{f}(\boldsymbol{x},\boldsymbol{y}) - \bar{\boldsymbol{f}}]^2 \sum_{x,y}[\boldsymbol{t}(\boldsymbol{x},\boldsymbol{y}) - \bar{\boldsymbol{t}}]^2\right)^{1/2}}$$

Ecuación 17 - Correlación normalizada

dónde:

- f(x,y) es la imagen.
- $\bar{t}$ es la media del patrón
- $\bar{f}$ es la media de la imagen en la región debajo de la plantilla.

En la Tabla 4 se pueden observar las funciones incluidas en Matlab que se han usado para esta parte.

Nombre de la función	Parámetros de entrada	Descripción	Parámetros de salida
I=rgb2gray(RGB)	RGB: imagen sobre la que se quiere realizar la conversión.	Convierte la imagen RGB a una imagen de intensidad en escala de grises.	I: imagen de salida.
C=normxcorr2(TEMPLATE,A)	TEMOLATE: plantilla para calcular la correlación. A: Imagen sobre la que trabajar.	Calcula la correlación cruzada normalizada de la matriz PLANTILLA y A.	C: matriz resultado, contiene los coeficientes de correlación y sus valores pueden oscilar entre -1.0 y 1.0.
[Y,Indx]=max(X)	X: vector de entrada.	Obtiene el valor máximo.	Y: valor máximo. Indx: posición en el array X del máximo encontrado.
[I,J]=ind2sub(SIZ,IND)	SIZ: tamaño de la matriz a la se quiere realizar la conversión. IND: vector de los índices que se quieren convertir.	Devuelve los arrays I y J que contiene la fila y la columna correspondiente a IND para una matriz de tamaño SIZ.	I, J: vectores correspondientes a la indexación múltiple.

Tabla 4 - Implementación prueba_corr

Con el siguiente ejemplo se van a explicar al detalle los pasos que hay que realizar a la hora de implementar un detector basado en la correlación normalizada. Se va utilizar el vídeo correspondiente al telediario del 14 de enero del 2012 de Televisión Española, se utiliza el frame 1051.

- SELECCIÓN DE LA PLANTILLA.

El usuario ha seleccionado el frame 951 del vídeo como origen de la plantilla. En este frame ha decidido destacar la parte correspondiente a la vestimenta de la presentadora, un vestido de color verde, pues es

una región que se diferencia bien del resto. El usuario aplica un recorte manual para quedarse con la característica que ha seleccionado.

Figura 29 - Selección de la plantilla y frame en estudio

- CORRELACIÓN NORMALIZADA Y BÚSQUEDA DE LAS COORDENADAS DEL MÁXIMO.

Nótese que la función `normxcorr2` solo funciona en imágenes en escala de grises, por lo que hay que realizar una transformación de la imagen previamente.

El sistema calcula la correlación cruzada normalizada, que se puede mostrar el en una gráfica de superficies 3D coloreada, Figura 30.
El pico, o máximo, se produce cuando el patrón y la imagen están mejor correlacionados.

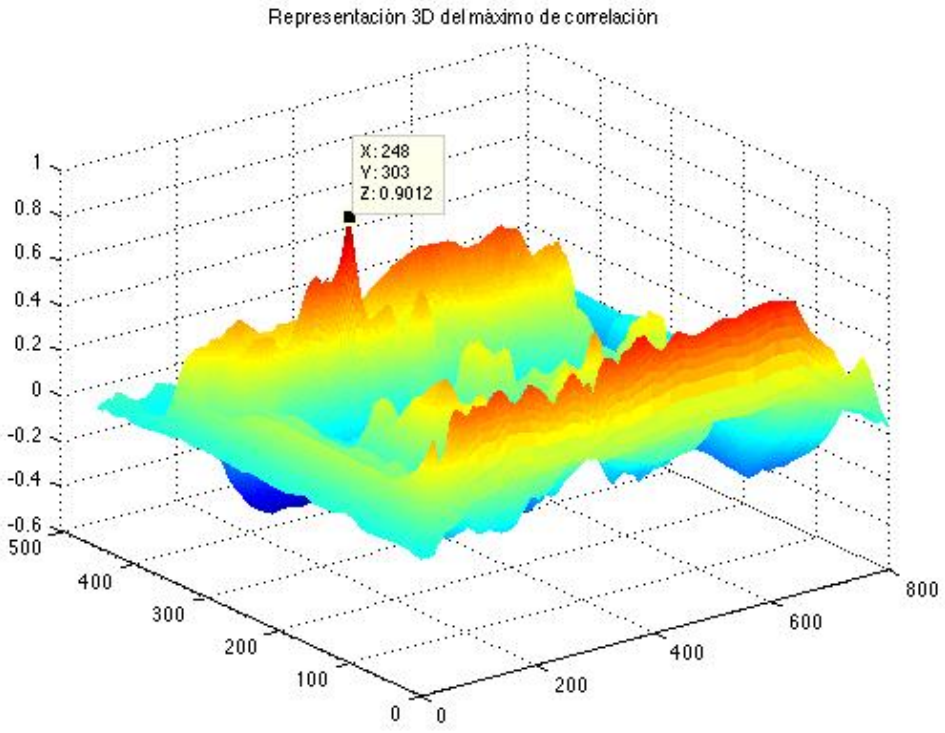


Figura 30 - Máximo de correlación 3D

Se obtiene `c`, matriz resultado de la correlación, que contiene los coeficientes de correlación. Para mostrar los elementos de `c` en el orden en el que se han ido almacenando se deben usar las siguientes líneas de código:

```
for n=1:size(c,2)
     for m=1:size(c,1)
        disp(c(m,n));
    end
 end
```

Para saber si la indexación múltiple que le corresponde al índice `imax`, se recuerda que es la posición del máximo de correlación en la matriz de correlación `c`, está bien realizada sólo se tiene que probar las siguientes líneas:

```
for q=1:1
    disp([c(ypeak(q),xpeak(q)), c(imax(q))]);
 end
```

Y se obtiene:

```
0.9012    0.9012
```

Para saber dónde se encuentra este máximo de correlación:

```
posicion=[ypeak(q), xpeak(q)]
```

Con lo que se obtiene la posición señalada con las coordenadas del punto inferior izquierda:

```
posicion =

   303   248
```

Figura 31 - Máximo de correlación (xpeak, ypeak)

Como los valores obtenidos se corresponden con los visualizados en la gráfica de superficies de `c` se da por válido el procedimiento.

Una mejora que se incluye en el sistema para reducir los fallos producidos por la correlación normalizada en el caso de no superar el umbral de correlación determinado por el usuario, fue incluir un bloque de escalado de la imagen patrón. De esta forma, se calcula la correlación para diferentes tamaños de la plantilla hasta obtener el resultado adecuado.

Nombre de la función	Parámetros de entrada	Descripción	Parámetros de salida
B = imresize(A, SCALE)	A: matriz sobre la que queremos realizar el recorte. SCALE: factor de escala.	Devuelve una imagen B que es la imagen A escalada un factor SCALE.	B: matriz escalada.

Tabla 5 - Implementación resize

- BÚSQUEDA DEL MÁXIMO DE CORRELACIÓN EN LA IMAGEN ORIGINAL.

  El punto de coincidencia máximo entre la imagen y el patrón, `corr_offset`, depende de la ubicación del pico en la matriz de correlación, `[xpeak ypeak]`, y del tamaño de la imagen con la que se está trabajando `[h w]`.

  ```
  corr_offset = [(xpeak-w) (ypeak-h)];
  ```

		Tamaño plantilla		corr_offset
xpeak	248	w	161	87
Ypeak	304	h	91	213

Figura 32 - Máximo de correlación (corr_offset[1] corr_offset[2])

- BÚSQUEDA DEL ÁREA DE COINCIDENCIA.

  Una vez que se tiene localizado el máximo de correspondencia en la imagen original se detecta el área que se encuentra bajo la plantilla para ese máximo, esta área corresponderá con el tamaño de la plantilla.

Figura 33 - Área de coincidencia

3.7. DETECCIÓN DE CORRESPONDENCIA

Como se deduce del apartado anterior, ya se tiene detectada cuál es el área bajo el patrón que corresponde al máximo de correlación encontrado. Se dibuja una matriz de ceros, a la que se denominará de ahora en adelante matriz de correlación, del mismo tamaño que la imagen con la que se esté trabajando y se ponen a uno los píxeles que equivalen al área de correlación.

Para el ejemplo en cuestión se tienen las siguientes plantillas, Figura 34.

Figura 34 - Plantillas

Se tendrán tantas matrices de correlación y tantos máximos de correlación como plantillas se hayan utilizado, siempre y cuando el máximo de correlación y el área de coincidencia se encuentren dentro de la imagen.

Como se puede observar en la imagen, para la primera plantilla no se ha hallado área de coincidencia pues en la posición a la que se encuentra no se puede encontrar un área igual al área de la plantilla que esté dentro de la imagen. Los valores del máximo de correlación son:

- Plantilla 1: máximo de correlación de 0.80.
- Plantilla 2: máximo de correlación de 0.68.

- Plantilla 3: máximo de correlación de 0.90.

Figura 35 - Plantillas

Se incluyen de forma ordenada todos los máximos obtenidos en un vector, vector de máximos. En la Figura 36 se muestra la situación de los puntos donde se ha dado el máximo de correlación.

Figura 36 - Posiciones de los máximos: rojo – plantilla 1, azul – plantilla 2, verde – plantilla 3

De todos los máximos obtenidos se escoge el mayor, pues corresponderá a la plantilla que más se parece a la imagen con la que se está trabajando. A partir de la posición de ese máximo en el vector de máximos se puede saber que matriz de correlación le corresponde y por lo tanto visualizar y estudiar el área que el sistema ha marcado como similar.

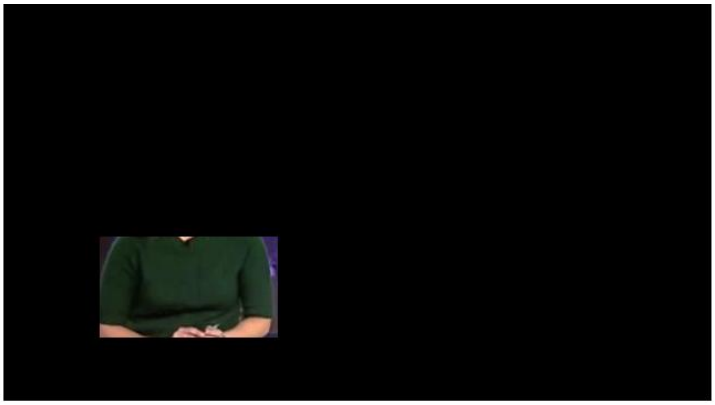


Figura 37 - Resultado final de la correlación

3.8. AMPLIACIÓN DEL SISTEMA

A la vista de los resultados obtenidos en las primeras pruebas a las que se sometió el sistema y que se recogen en el capítulo 4 se notó que no eran tan precisos como se esperaba.

Se introdujo un nuevo bloque en el sistema que fuese capaz de obtener una nueva plantilla de forma automática. Esta plantilla, que se centra en la cara de la persona en estudio, se obtiene a raíz de una serie de métodos que utilizan como base la primera plantilla que el usuario selecciona. Por lo que se parte de la premisa de que el usuario ha tenido que escoger una primera plantilla en la que se pueda ver la región de la cara de la persona.

Para este procedimiento se llevaron a cabo dos procesos: en primer lugar distinguir los píxeles de la piel y en segundo lugar seleccionar el bloque de pixeles que pertenecen a la cara.

- DETECCIÓN DE PÍXELES DE PIEL

A la hora de construir una regla que sea capaz de diferenciar los pixeles correspondientes a la piel dentro de una imagen se han encontrado diferentes metodologías [6] [13]:

- Modelos no paramétricos de distribución de la piel, empleados en [32]. La idea clave consiste en estimar la distribución del color de la piel por medio del entrenamiento, sin determinar un modelo de color en concreto.

- Modelos paramétricos de distribución de la piel [29], cuyos métodos trabajan en el plano del modelo de color de la crominancia, obviando toda la información que puede contribuir la luminancia.

- Métodos de distribución de piel dinámicos [25], métodos auto-actualizados para generar modelos específicos en función de la

distribución del color de piel que puede verse influenciada por cambios en las condiciones de iluminación.

- Definiciones explícitas de regiones de color en los diferentes espacios.

Para el desarrollo del presente trabajo, se quería construir un clasificador de píxeles de piel rápido y dado que detectar los píxeles de piel no era un objetivo esencial se emplearon las definiciones explícitas de regiones de color. Se tuvieron en cuenta los siguientes antecedentes:

- Estudiando la documentación proporcionada en [23] los modelos de color que ofrecen mejores resultados en esta materia son Xerox/YES con una probabilidad de acierto del 95%, YIQ con un 80% de aciertos e YC_bC_r con un 79%.

- En [6], de los modelos estudiados (RGB, HSV e YC_bC_r), es el segundo el que ofrece mejores resultados.

- En [17] se explica que a la hora de detectar pixeles de piel en el espacio de color YC_bC_r para personas no caucásicas, la componente correspondiente al valor de la luminancia, Y, es muy diferente, por lo que sólo se tienen en cuenta los valores de las componentes de crominancia, C_b y C_r. En cambio, al usar el método HSV para los diferentes canales los histogramas se asemejan más independientemente de la raza.

- En [13] se expone que no es eficiente tener en cuenta la componente de la luminancia, pues el color de la piel está descrito por las crominancias, por lo que el modelo RGB no sería el más eficiente aun obteniendo resultados aceptables.

Finalmente para este trabajo se ha optado por la utilización del espacio de color HSV, pues como se ya se ha comentado el rango de tonalidades de piel es mucho más amplio sin descartar ninguno de los canales, y por lo tanto sin perder información.

El sistema de detección de píxeles de piel consiste en recorrer toda la imagen y para cada píxel, comprobar que su valor esté dentro de unos márgenes. Si es así, el píxel será marcado como píxel de piel y en la imagen resultante de esta sección se percibirá como un pixel blanco. Las condiciones que debe cumplir para determinarlo como tal según son [18]:

$$0 < H \leq 0.12$$
$$0.5 < S < 0.9$$
$$0.2 < V < 0.95$$

Una vez que tenemos marcados los píxeles que pertenecen a la cara, se procedió con el empleo de una operación morfológica, la apertura. Gracias a esta operación se suavizan los contornos de la imagen, se eliminan pequeñas protuberancias y se rompen las conexiones débiles. En la Figura 38 podemos ver un ejemplo de esta operación morfológica.

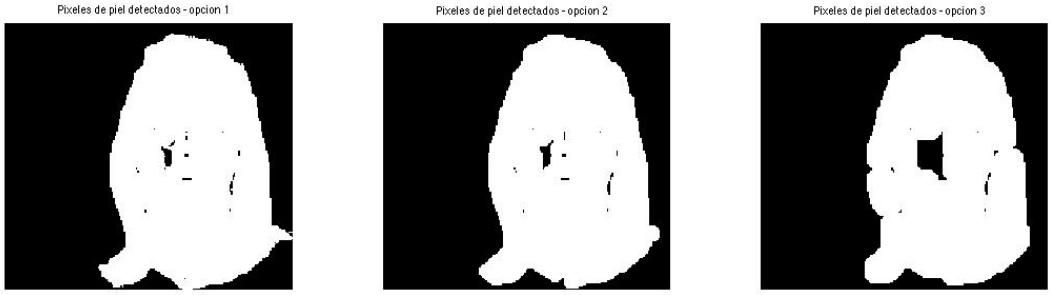


Figura 38 - Opción 1- detección de pixeles de la piel sin apertura; Opción 2- detección de pixeles de la piel con apertura, radio 5; Opción 3- detección de píxeles de piel con apertura, radio 10.

El elemento estructurante es de tipo *disk* con radio 3, Figura 39 . Se eligió este valor pues, a raíz de pruebas realizadas, es el tamaño de disco más pequeño para el cual se suavizan los bordes y se eliminan protuberancias sin deformar en exceso la imagen.

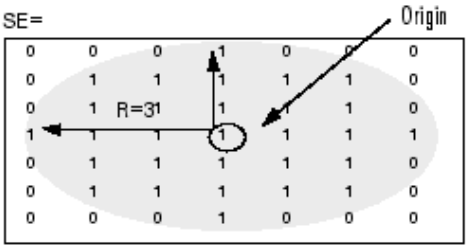


Figura 39 - Elemento estructurante de tipo disco con radio 3.

Se utilizan las funciones que se explican en la Tabla 6 tanto para describir el elemento estructurante como para realizar la operación de apertura.

Nombre de la función	Parámetros de entrada	Descripción	Parámetros de salida
se=strel('disk',R)	'disk': string que define la forma del elemento estructurante. R: radio.	Crea un elemento estructurante con la forma definida por el primer parámetro y el tamaño definido en el segundo.	se: elemento estructurante.
IM2=imopen(IM,se)	IM: matriz sobre la que se quiere realizar la operación morfológica. se: elemento estructurante.	Realiza la operación morfológica de apertura con el elemento estructurante se.	IM2: matriz de salida.

Tabla 6 - Implementación pielhumana_hsv

- DETECCIÓN DE LOS PÍXELES DE LA CARA

Cuando ya se tienen localizados los pixeles de la piel, aquellos que son blancos, se procede a extraer únicamente los correspondientes a la cara.

Como ya se está trabajado con una imagen en escala de grises el siguiente paso es binarizarla con el umbral que tiene. Para describir las regiones se usa la función bwlabel, que convierte la imagen en una matriz de etiquetas. Se toma esa matriz etiquetada y se obtiene las propiedades que se deseen. En nuestro caso se escogieron:

- Area, que calcula el área en pixeles cuadrados de la región.
- Boundingbox, calcula la posición y dimensiones del mínimo rectángulo que envuelve la región.
- Centroid, calcula la posición del centroide de la región.

Observando los vídeos de entrenamiento se notó que la cara era la mayor región de píxeles de la piel que se ve en los frames, por lo que se implementaron un conjunto de funciones que devolvieran esta región.

Esta parte del código se encuentra implementada en la función imgBox cuyas funciones principales se explican a continuación.

Nombre de la función	**Parámetros de entrada**	**Descripción**	**Parámetros de salida**
level=graythresh(I)	I: matriz sobre la que queremos trabajar.	Realiza una umbralización de la imagen usando el método Otsu.	level: valor de intensidad normalizado, en el rango de [0,1].
BW=im2bw(I, level)	I: matriz sobre la que queremos trabajar. Level: umbral para realizar la binarización.	Convierte la imagen en escala de grises I a una imagen binaria. Los píxeles de entrada que tengan un valor de luminancia mayor que level pasan a valer 1 (blanco) mientras que el resto vale 0 (negro).	BW: imagen binaria
[L,NUM]=bwlabel(BW,N)	BW: matriz	Etiquetado de	L: matriz

	binaria con la que se quiere trabajar. N: especifica el número de objetos conectados, por defecto vale 8.	objetos en una imagen binaria	resultado que contiene las etiquetas de los objetos encontrados. NUM: número de objetos conectados encontrados en BW.
STATS=regionprops(BW,PROPERTIES)	L: matriz con la que se quiere trabajar. PROPERTIES: lista de medidas que se quieren tomar, en nuestro caso ‘BoundingBox', 'Area' y 'Centroid'	Calcula las propiedades de las regiones etiquetadas	STATS: estructura que contiene las medidas para cada propiedad de cada región etiquetada.
Y=sort(X,MODE)	X: vector que se quiere ordenar. MODE: dirección de ordenación (ascendente o descendente)	Ordena el array	Y: array ordenado

Tabla 7 - Implementación imgBox

La Figura 40 muestra un ejemplo de todo este proceso.

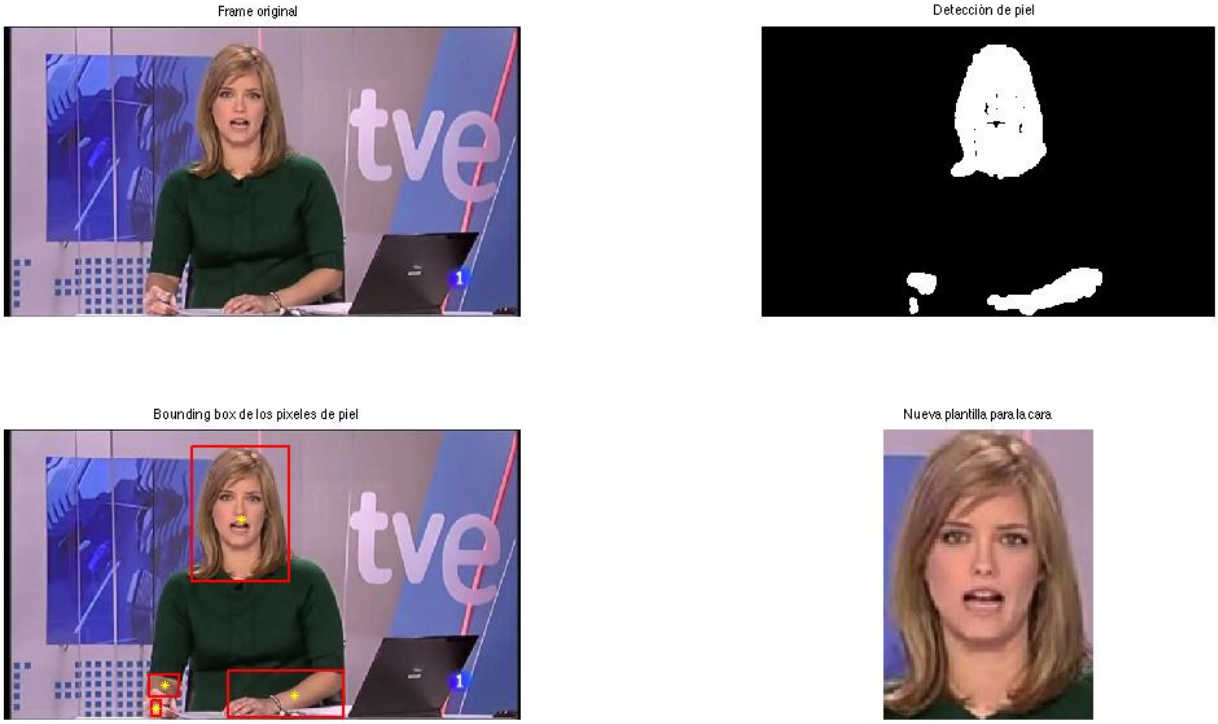


Figura 40 - Obtención de una nueva plantilla

Capítulo 4
Evaluación del sistema

En la sección 4.1 de este capítulo se van a analizar los vídeos con los que se han realizado las pruebas pertinentes para valorar el sistema. Además de una breve descripción de lo que se puede apreciar en los fragmentos utilizados se incluyen algunos de los frames principales así como sus datos técnicos y de interés para el trabajo.

Posteriormente se presentan y analizan los resultados obtenidos por el sistema desarrollado para la detección de personas en secuencias de vídeo. El objetivo principal del proyecto no es conseguir un sistema absolutamente fiable, sino investigar el método de *template matching* a través de la correlación e intentar optimizar este sistema para una casuística determinada.

En la sección 4.2 se describen los resultados en base a la matriz de confusión además se incluye una descripción cualitativa de los mismos y en la sección 4.3 se analizan algunos casos específicos.

4.1. CORPUS DE LOS VÍDEOS.

Se han escogido vídeos de diferente naturaleza para realizar unas pruebas lo más dispares posibles. Que cada vídeo tenga una calidad diferente sirve para realizar las pruebas con diferentes umbrales de correlación: si la calidad es baja el umbral deberá ser bajo en cambio, si la calidad del vídeo es alta el usuario puede hacer que el sistema sea más fino y determinar un umbral alto también.

- Telediario del 10 de enero del 2011, Antena 3.

Este vídeo se corresponde con los primeros minutos del telediario emitido la noche del 10 de enero del 2011 en la cadena Antena 3. En él se pueden apreciar varios personajes, porque hay un momento en el que la cámara realiza un barrido general y muestra hasta los técnicos del plató, pero se podría decir que en primer plano solo se cuentan 3 personajes.

Frame 276	Frame 476	Frame 1201	Frame 1326

Tabla 8 – Telediario del 10 de enero del 2011, Antena 3

- Telediario del 14 de enero del 2012, Televisión Española.

Refleja los últimos minutos del telediario emitido el 14 de enero del 2012 por la cadena Televisión Española. Únicamente se pueden apreciar tres

personajes principales que se encuentran durante todo el vídeo en la misma posición, dos de ellos a menudo aparecen en el mismo plano.

Frame 26	Frame 451	Frame 576	Frame 901

Tabla 9 – Telediario del 14 de enero del 2012, Televisión Española

- "Pesadilla en la cocina", Antena 3.

Este programa fue emitido la noche del 20 de junio del 2013 por la cadena La Sexta. En el fragmento del vídeo a analizar podemos ver a dos de sus protagonistas desplazándose por una cocina, por lo que su posición frame tras frame nunca es la misma.

Frame 101	Frame 601	Frame 876	Frame 1276

Tabla 10 – "Pesadilla en la cocina", Antena 3

- "Previo GP Italia 2013", Antena 3.

Se muestra la entrevista que realiza el presentador a un piloto de Fórmula 1 los días previos a la carrera, 8 de septiembre del 2013. Los personajes permanecen siempre en la misma posición y los planos de cámara son cortos.

Frame 30	Frame 1016	Frame 1799	Frame 2727

Tabla 11 – "Previo GP Italia 2013", Antena 3

- "El hormiguero", Antena 3.

Programa emitido la noche del 28 de noviembre del 2013, en el que se puede ver al presentador realizando una entrevista a dos invitados. Todos los personajes permanecen en una posición estática durante todo el vídeo, aunque la cámara va enfocando a la persona que se encuentra hablando en cada momento.

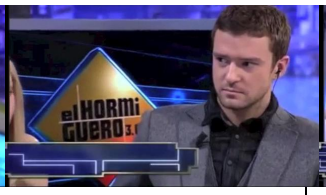
			
Frame 401	Frame 901	Frame1601	Frame 1701

Tabla 12 – "El hormiguero", Antena 3

En la tabla que se muestra a continuación se pueden apreciar los datos técnicos de los vídeos utilizados para las pruebas.

Nombre del vídeo	**Duración**	**Frames**	***Frames per second***	**Número de personajes principales**	**Calidad**
Antena3.avi	01:17	78	25	varios	Media-baja
TelevisionEspanola.avi	00:49	49	25	3	Media-alta
PesadillaEnLaCocina.avi	01:00	61	25	2	Media-baja
PrevioAntena3.avi	01:34	98	30	2	Media-alta
ElHormiguero.avi	01:12	73	25	3	Media-alta

Tabla 13 – Datos técnicos de los vídeos de prueba

4.2. MATRIZ DE CONFUSIÓN

En el contexto de las tareas de clasificación, para cuantificar el rendimiento y la precisión del sistema desarrollado se pueden contabilizar

los frames donde éste acierta o falla, utilizando los siguientes parámetros [7]:

- TP, *true positive*: Eventos detectados correctamente.
  - En este trabajo implica que se ha detectado la plantilla correctamente en un frame.

- FN, *false negative*: Eventos no detectados.
  - Se refiere a las situaciones en las que el sistema no es capaz de detectar la plantilla en un frame.

- FP, *false positive*: Eventos que el sistema detecta pero no son correctos.
  - Situaciones en las que el sistema detecta algo que no es la plantilla.

- TN, *true negative*: Eventos correctamente rechazados.
  - Cuando el sistema no tiene que detectar nada en un frame y así lo hace.

La Figura 41 [5] muestra la matriz de confusión que representa los parámetros explicados anteriormente.

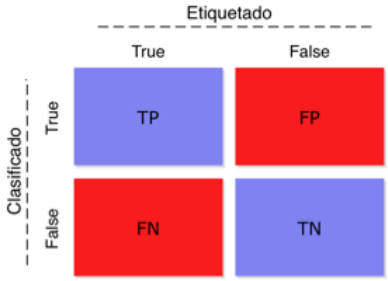


Figura 41 - Representación gráfica de la matriz de confusión.

A partir de ellos se pueden extraer las siguientes ecuaciones que representan los ratios más comunes [7]:

- TPR, *true positive rate,* probabilidad de detección ($P_{detección}$) o *recall*: Mide que las instancias de la clase C se clasifiquen como C, aunque otras instancias también se clasifiquen como clase C sin serlo.
  - Es el número de elementos positivos detectados (TP), entre el total de los que realmente pertenecen a la clase positiva (TP + FN).

$$Recall = P_{detección} = \frac{TP}{TP + FN}$$

- FPR, *false positive rate*: Equivale a la probabilidad de falsa alarma ($P_{falsa\ alarma}$), la probabilidad de que cuando un evento no pertenezca a la clase C, el sistema lo identifique como perteneciente a la clase C.
  - Es el número de eventos que el sistema detecta incorrectamente (FP), entre el total de los elementos que pertenecen a la clase negativa (TN + FP).

$$P_{falsa\ alarma} = \frac{FP}{TN + FP}$$

- Precisión: Mide que las instancias de clasificadas como pertenecientes a la clase C sean realmente de la clase C, aunque haya instancias de esta clase que se clasifiquen como otra clase.
  - Es el número de elementos positivos (TP), entre todos los eventos detectados como positivos (TP+FP).

$$Precisión = \frac{TP}{TP + FP}$$

- **FASE 1 DE PRUEBAS**

Una vez que se tuvo todo el código desarrollado, la primera prueba a la que se sometió el sistema fue enfrentarse al problema de la detección trabajando con una única plantilla. Los resultados obtenidos se muestran en la tabla siguiente.

		TP	FN	FP	TN	Recall o $P_{detección}$	$P_{falsa\ alarma}$	Precisión
1	**Telediario 10 de enero del 2011, Antena 3**	5	38	1	33	**0.12**	**0.02**	**0.83**
2	**Telediario 14 de enero del 2012, Televisión Española**	12	25	0	11	**0.32**	**0**	**1**
3	**"Pesadilla En La Cocina", Antena 3**	4	39	3	14	**0.09**	**0.18**	**0.57**
4	**"Previo F1 Italia", La Sexta**	12	79	6	0	**0.13**	**1**	**0.66**
5	**"El Hormiguero", Antena 3**	6	48	3	15	**0.11**	**0.17**	**0.66**
	Precisión media							**0.74**

Tabla 14 – Resultados de fase 1, prueba 1

Para entender mejor los resultados expuestos anteriormente se ha incluido la gráfica precisión-*recall* que representa la precisión del algoritmo contra la cobertura del mismo.

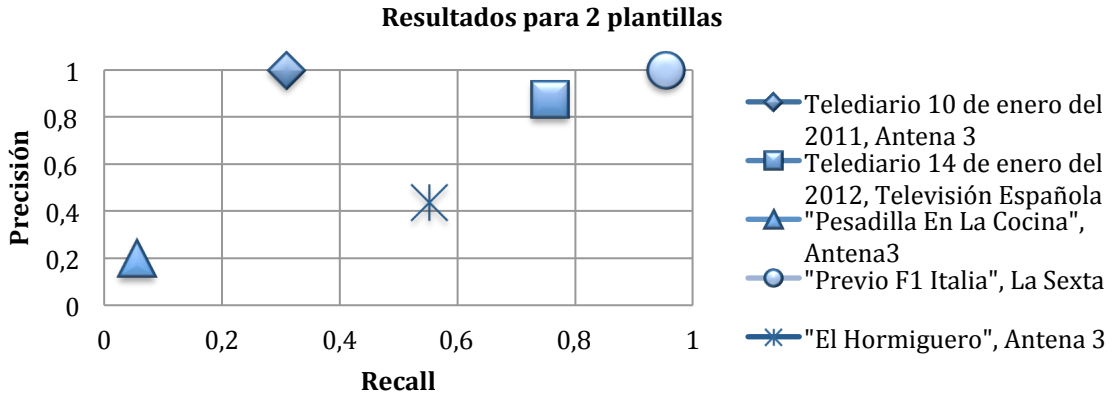
Como se puede observar, los resultados no son todo lo buenos que deberían pues la cobertura del sistema es muy baja, por lo que se volvieron a realizar las mismas pruebas ampliando el número de plantillas.

Para este segundo caso, se siguió trabajando con las plantillas de la primera prueba a las que se sumó una nueva plantilla que se correspondía con la vestimenta de la persona en estudio. Los resultados obtenidos se muestran a continuación.

		TP	FN	FP	TN	Recall o $P_{detección}$	$P_{falsa\ alarma}$	Precisión
1	**Telediario 10 de enero del 2011, Antena 3**	13	29	0	34	**0.30**	**0**	**1**
2	**Telediario 14 de enero del 2012, Televisión Española**	28	9	4	6	**0.75**	**0.40**	**0.875**
3	**“Pesadilla En La Cocina”, Antena 3**	2	34	8	15	**0.05**	**0.35**	**0.20**
4	**“Previo F1 Italia”, La Sexta**	41	2	0	53	**0.95**	**0**	**1**
5	**“El Hormiguero”, Antena 3**	21	17	27	6	**0.55**	**0.82**	**0.44**
	Precisión media							**0.70**

Tabla 15 – Resultados de fase 1, prueba 2

Se incluye también la gráfica precisión-*recall* para visualizar mejor los resultados.



- **FASE 2 DE PRUEBAS**

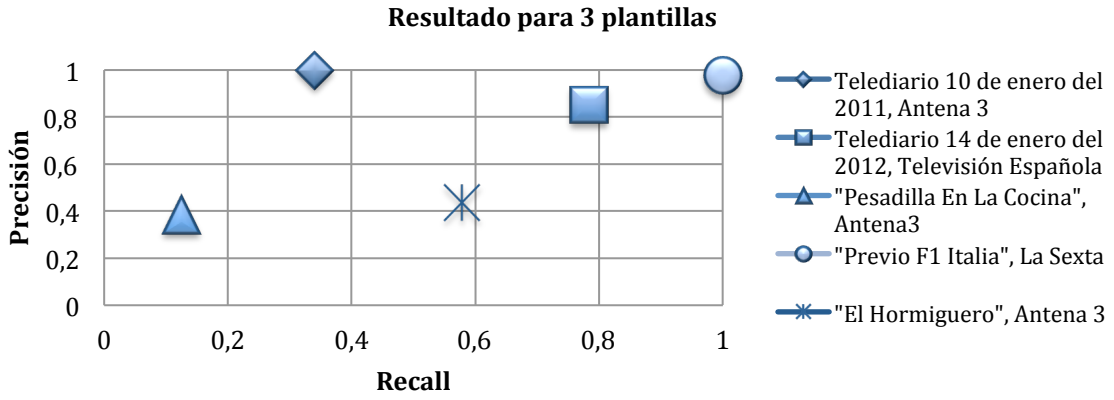
Como ya se comentó en el capítulo 3 Diseño e Implementación, se quiso perfeccionar el sistema para que el algoritmo de detección fuese lo más preciso posible. Los siguientes datos muestran los resultados obtenidos cuando se incluye el bloque de detección de la cara y su correspondiente plantilla.

El experimento se realizó con las mismas plantillas que las pruebas de la fase 1 pero incluyendo una plantilla más que se centrase en la cara de la persona en estudio. Los resultados obtenidos se muestran a continuación.

		TP	FN	FP	TN	Recall o $P_{detección}$	$P_{falsa\ alarma}$	Precisión
1	**Telediario 10 de enero del 2011, Antena 3**	13	29	0	34	**0.30**	**0**	**1**
2	**Telediario 14 de enero del 2012, Televisión Española**	27	8	5	7	**0.77**	**0.42**	**0.84**
3	**"Pesadilla En La Cocina", Antena 3**	3	35	8	13	**0.07**	**0.38**	**0.27**
4	**"Previo F1 Italia", La Sexta**	41	0	1	54	**1**	**0.02**	**0.98**
5	**"El Hormiguero", Antena 3**	22	16	24	9	**0.58**	**0.73**	**0.48**
	Precisión media							**0.71**

Tabla 16 – Resultados de fase 2, prueba 1

La gráfica precisión-*recall* muestra los resultados.



- **ÁNALISIS DE LOS RESULTADOS**

Observando los resultados recogidos en los apartados anteriores se pueden percibir los siguientes sucesos:

- Comparando los resultados obtenidos en la fase 1, se ve que a mayor número de plantillas la $P_{detección}$ o *recall* aumenta, con una plantilla el valor medio de la $P_{detección}$ es igual a 0.15 mientras que para dos plantillas es 0.52. Esto es debido a que ampliando el número de plantillas se juega con más patrones con los que realizar la detección, por lo que la probabilidad de encontrar alguna similitud aumenta y por lo tanto aumentan los TP.

- También hay que tener en cuenta la implicación del uso de más plantillas en el caso contrario. Al tener más patrones la probabilidad de confundir una región del frame en estudio con un patrón es mayor, lo que provoca que aumenten los FP y por lo tanto disminuya la tasa de precisión. Se puede comprobar que para el caso con menor número de plantillas se obtiene la precisión más alta, con un valor de 0.74.

- La $P_{falsa\ alarma}$ representa la siguiente casuística: bajo la hipótesis de que el patrón no aparece en el frame en estudio, el sistema si encuentra una correlación entre la plantilla y el frame mayor que el umbral, por lo tanto si detecta la correspondencia entre las imágenes. Esta probabilidad es mayor según vamos aumentando la cantidad de plantillas con las que trabajamos. Por eso, para la primera prueba de la fase 1 se ha obtenido el menor valor medio de esta probabilidad, 0.274; mientras que para la segunda prueba de esta fase y la primera prueba de la fase 2 se han obtenido un valor en media aproximadamente de 0.31 para ambos casos.

- Como ya se ha dicho, a mayor número de plantillas con las que entrenar el sistema el valor del *recall* aumenta. Esto también se puede ver

reflejado si se compara el caso de 2 plantillas con el caso de 3 plantillas. El primero de ellos tiene una $P_{detección}$ de 0.52 mientras que el segundo de 0.55.

4.3. ESTUDIO DE CASOS ESPECÍFICOS

Observando los resultados obtenidos de la primera prueba realizada, aquella que únicamente empleaba una imagen de la cara como plantilla para realizar la correlación, se puede decir que el número de eventos no detectados (que se corresponde con FP) es muy elevando, aproximadamente más del 50%, mientras que muy pocos se detectan correctamente. Se debe a que con una sola plantilla que represente a la persona en estudio el sistema no es capaz de obtener una correlación que supere el umbral marcado. Los frames en estudio tienen que contener una región que se parezca mucho al patrón para detectar correctamente la similitud.

			
Plantilla	Frame 276	Frame 776	Frame 1376

Tabla 17 – *True positive* de fase 1 prueba 1 para el vídeo "El Hormiguero"

La Tabla 17 refleja que los vídeos detectados correctamente como TP se parecen demasiado al patrón. En la Tabla 18 podemos ver como otros frames que también se parecen no obtiene una correlación que supere el umbral de 0.8, por lo que se clasifican como eventos no detectados

			
Plantilla	Frame 1001	Frame 1176	Frame 1226

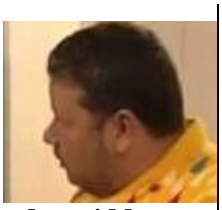
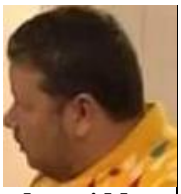
Tabla 18 – *False negative* de fase 1 prueba 1 para el vídeo "El Hormiguero"

Para el caso del frame 1226, destacar que al incluir la plantilla de la ropa (fase 1 prueba 2) se obtiene una correlación del 0.8030, mientras que en la fase 2 prueba 1 la plantilla que se centra en la cara no consigue superar de nuevo el límite, quedándose con una similitud de 0.7552.

	 Plantilla 1	 Plantilla 2	 Plantilla 3
 Frame 1226	0	0.7552	0.8030

Tabla 19 – Correlación de las diferentes plantillas para el vídeo "El Hormiguero"

Cuando tenemos un vídeo de peor calidad y en el que la persona en estudio se encuentra en constante movimiento, como puede ser "Pesadilla En La Cocina", aún con el mayor número de plantillas que se ha probado, es muy difícil que el sistema obtenga buenos resultados. Es debido a que en muy pocos frames la persona mantiene su posición. Además, cuanto peor es la calidad más se notan los cambios de iluminación y la baja definición de la imagen. Para solucionar este problema habría que bajar el umbral hasta un valor alrededor de 0.7.

	 Plantilla 1	 Plantilla 2	 Plantilla 3
 Frame 76	0.7732	0.7439	0.7340
 Frame 676	0	0.7307	0.6849

 Frame 1501	0.6985	0.7338	0.6657

Tabla 20 – Correlación de las diferentes plantillas para el vídeo "Pesadilla En La Cocina"

Los vídeos de mayor calidad son en los que mejores resultados se obtienen. Además de porque la definición del imagen es óptima hay que destacar que en estos vídeos la persona en estudio se encuentra en la misma posición, lo que implica que la iluminación y las sombras no cambien de un frame a otro. Para estos vídeos se podía aumentar el umbral de correlación, porque cuando el sistema detecta la similitud entre la imagen y la plantilla, ésta tiene un valor muy elevado. Se pueden comprobar todos estos hechos con el vídeo "Previo F1 Italia" que tiene el mayor valor de $P_{detección}$.

	 Plantilla 1	 Plantilla 2	 Plantilla 3
 Frame 1	0.9723	0.9710	0.9902
 Frame 1219	0.8624	0.8808	0.8282
 Frame 1625	0.9909	0.98834	0.9293

Tabla 21 – Correlación de las diferentes plantillas para el vídeo "Previo F1 Italia"

Como se puede ver en la

Tabla 21 el uso de una plantilla específica para la cara en vídeos de muy buena calidad y estáticos es redundante porque con el empleo de dos plantillas sería más que suficiente.

Uno de los problemas que se ha detectado durante la realización de las pruebas es que en ocasiones el algoritmo es capaz de detectar una coincidencia entre la plantilla y la imagen cuando realmente no se da. Esto es así porque al realizar el *resize* de la plantilla la imagen es tan pequeña que se pueden confundir unos píxeles con otros. Para solucionar este problema se podría hacer al sistema más preciso, aumentando el umbral de correlación. A continuación se muestran algunas imágenes que ejemplifican este problema.

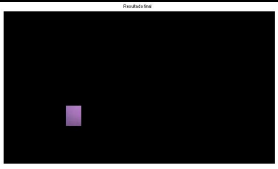
 Plantilla 2	 Frame 813	 Correlación de 0.8205
 Plantilla 1	 Frame 526	 Correlación de 0.8145
 Plantilla 1	 Frame 601	 Correlación de 0.8036

Tabla 22 – *False positive* para los vídeos "Previo F1 Italia", "El Hormiguero", y el telediario de la noche del 14 de enero del 2012

Capítulo 5

Conclusiones y líneas futuras

Capítulo 5

CLONCLUSIONES Y LÍNEAS FUTURAS

En el presente trabajo se han definido una serie de algoritmos a través de los cuales se puede llevar experimentos con el objetivo de detectar personas en secuencias de vídeo.

En este capítulo se revisan los resultados obtenidos con el fin de extraer conclusiones. Seguidamente se proponen algunas líneas futuras de trabajo entorno al sistema ya desarrollado.

5.1. CONCLUSIONES

Teniendo en cuenta la documentación estudiada antes de la implementación del sistema se notó que el método seleccionado no era el que mejor resultados aportaba. La correlación, aunque para ciertos casos si obtiene muy buenos resultados y no supone un proceso de implementación excesivamente complicado, conlleva un elevado coste computacional que puede hacer minimizar sus ventajas.

Con el sistema desarrollado y en base a las pruebas realizadas se podría decir que es un método especialmente válido para secuencias de personas en las que no se aprecia mucho movimiento de cámara y que tiene unas condiciones de iluminación constantes, como pueden ser los telediarios o las entrevistas. Para el resto de casos, el método de la correlación no es tan preciso como se quiera aun ampliando considerablemente el número de plantillas con las que trabajar. Además, el uso de varias plantillas perjudica al tiempo de ejecución considerablemente y no siempre resulta beneficioso, porque como se puede ver en las pruebas realizas a veces las plantillas son redundantes.

Si lo que se quiere es trasladar este sistema al día a día y utilizar este trabajo en un entorno real hay que tener en cuenta varios factores:

- Resultados obtenidos. Aunque en cómputo global se podría decir que los resultados obtenidos son aceptables hay que destacar que en los mismos influye mucha la selección de las plantillas por parte del usuario. Si éste no es capaz de delimitar cuales son las características de interés y más destacables de la persona en estudio el sistema puede no ser tan preciso como en las pruebas reflejadas en el capítulo 4.

- Tiempo de ejecución. La correlación cruzada normalizada es una operación del dominio del tratamiento de imágenes que conlleva un coste computacional muy elevado si se utilizan muchas plantillas y/o plantillas de amplias dimensiones. Hay que mantener una relación correcta entre el tiempo de ejecución y el número de plantillas que se utilizan, pues no por usar más cantidad de plantillas se obtienen mejores resultados.

- También hay que tener en cuenta el factor económico. El coste del desarrollo de sistema tiene que ser proporcional a los resultados favorables conseguidos.

5.2. LÍNEAS FUTURAS

A la vista de los resultados obtenidos han surgido varias líneas para continuar con las técnicas usadas en este trabajo.

- Bases de datos

Una posible mejora sería emplear bases de datos estandarizadas para así poder comparar fielmente este algoritmo con otros algoritmos desarrollados para la misma tarea.

- Preprocesado

Se podrían usar más técnicas de preprocesado que beneficiasen al sistema en la tarea de la detección. Esto beneficiaría sobre todo a los vídeos de baja calidad.

- Ajuste de la orientación de la plantilla

De la misma forma que se implementa un método para trabajar con diferentes tamaños de la plantilla se puede desarrollar otro bloque que sirva para rotar las imágenes. De esta forma si en una imagen la persona en estudio inclina la cabeza el sistema sería capaz de detectarlo.

- Empleo de características SIFT

Scale invariant feature transform [21] es un algoritmo que permite obtener características relevantes a objetos de una escena no segmentada con invariancia de la posición, escala y rotación que posteriormente pueden utilizarse en la detección de movimiento o en el reconocimiento de objetos.

- Almacenamiento final de las imágenes

Una vez que se tiene el frame clasificado se podría crear un sistema de almacenamiento para los que tengan un resultado favorable, de tal manera que se tengan los frames correctos almacenados y localizados de forma independiente.

Capítulo 6

Planificación y presupuesto

Capítulo 6

PLANIFICACIÓN Y PRESUPUESTO

Este capítulo recoge la planificación y el presupuesto de las tareas relacionadas con la implementación de un sistema de detección de personas en secuencias de vídeo.

6.1. PLANIFICACIÓN DEL TRABAJO

El trabajo consiste en la implementación de un sistema capaz de detectar personas en secuencias de vídeo. Para la ejecución de proyecto se han tenido en cuenta las siguientes fases.

- Adquisición de conocimientos. En esta fase se recopila y asimila toda la información referente a la detección de objetivos y se extrapola a personas. Está formada por las tareas de toma de contacto y búsqueda de referencias bibliográficas.

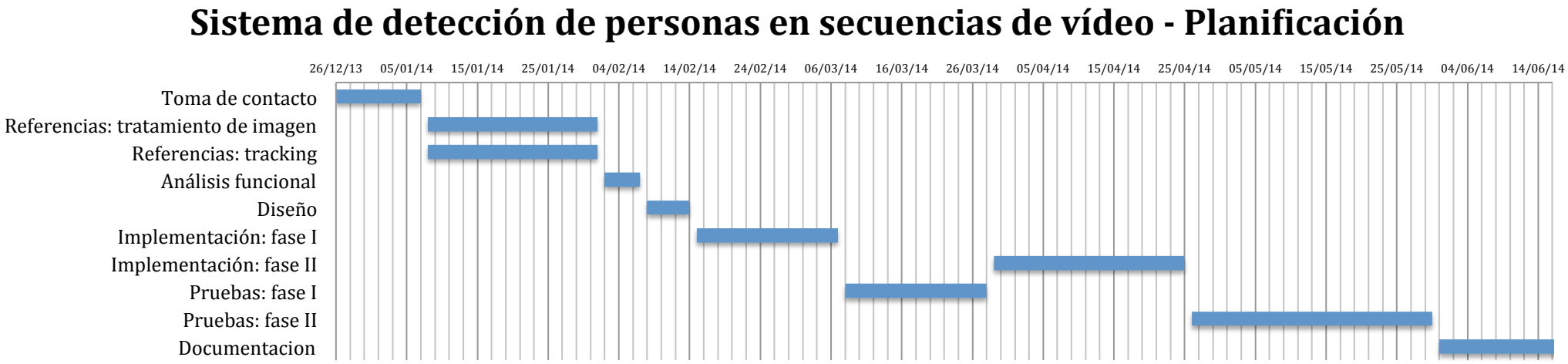
- Análisis. Durante esta fase se realiza el análisis funcional y de requisitos del sistema. También se diseña el diagrama que posteriormente será implementado.

- Implementación. Se realiza la implementación del sistema en dos etapas.

- Pruebas. Se realizan las pruebas pertinentes para valorar el sistema.

- Documentación. Se redacta la documentación que acompaña al trabajo.

En la siguiente figura se puede ver un diagrama de planificación, diagrama de Gantt, cuyo objetivo es exponer el tiempo de dedicación pronosticado para cada tarea.



	Documentacion	Pruebas: fase II	Pruebas: fase I	Implementación: fase II	Implementación: fase I	Diseño	Análisis funcional	Referencias: tracking	Referencias: tratamiento de imagen	Toma de contacto
FECHA DE INICIO	31/05/14	26/04/14	08/03/14	29/03/14	15/02/14	08/02/14	02/02/14	08/01/14	08/01/14	26/12/13
DURACION	17	34	20	27	20	6	5	24	24	12

Figura 42 Diagrama de Gantt

6.2. RESUMEN DE ROLES Y COSTES DEL PERSONAL

Para el desarrollo del trabajo se tendrán en cuenta los siguientes perfiles:

- 1 Ingeniero Sénior (tutor). Ingeniero de telecomunicación experto en Sistemas Inteligentes. Actuación como consultor funcional y responsable de la supervisión del trabajo y de la revisión de la documentación asociada.

- 1 Analista Programador (alumno). Ingeniero de telecomunicación con conocimientos en Matlab. Recopilación de información sobre sistemas de detección y el estado del arte. Desarrollo por completo del sistema y realización de las pruebas. Realización de la documentación asociada.

Para calcular los costes del personal asociados a este trabajo es necesario tener en cuenta la duración y las horas del mismo.

- El tiempo empleado por el alumno para la realización de este trabajo ha sido aproximadamente de 6 meses, 20 horas a la semana, por tanto el número total de horas ha sido de 480 horas. Los honorarios son de 33 €/hora, dando lugar a 15840 €.

- En cuanto al tutor se han considerado unos costes de 3500€, resultado de 40 horas a 70 €/hora.

Para el precio por hora y por perfil se ha tenido en cuenta una ponderación basada en las tarifas de varias consultoras tecnológicas para los perfiles de Analista Programador y Consultor Sénior durante el año 2014. El coste total de los recursos humanos dedicados al trabajo se muestra en la siguiente tabla.

Rol	Honorario	Horas	Total
Ingeniero Junior: Alejandra Benavides Blanco	33 €/hora	480	15840 €
Ingeniero Sénior: Julio Villena Román	70 €/hora	40	2800 €
TOTAL			**18640 €**

Tabla 23 – Costes recursos humanos

6.3. COSTES MATERIALES

Los costes materiales asociados a este proyecto han sido los siguientes:

- Ordenadores: El equipo personal del Ingeniero Junior se emplea para la programación de los diferentes algoritmos, y la redacción de la documentación del trabajo. El equipo empleado tiene un coste de 1229 €. Por otro lado, el Ingeniero Sénior emplea otro equipo para la revisión del trabajo, con un coste de 700 €.

  Para calcular el coste de los ordenadores imputable a la realización de este trabajo se ha utilizado la siguiente fórmula de amortización.

$$\frac{A}{B} \cdot C \cdot D \; donde \begin{cases} A: número\ de\ meses\ desde\ la\ fecha\ de\ facturación \\ B: periodo\ de\ depreciación \\ C: coste \sin I.V.A. \\ D: \%\ del\ uso\ que\ se\ le\ dedica \end{cases}$$

Ecuación 18 – Cálculo de la amortización

Equipo personal	Coste (sin IVA)	Uso dedicado al proyecto	Dedicación	Periodo de depreciación	Coste imputable
Portátil Ing. Junior (MacBook Pro Core i5 de Intel de doble núcleo a 2.5 GHz)	1016 €	90 %	6 meses	60 meses	203.2 €
Portátil Ing. Sénior (Dell Intel Core i3 de doble núcleo a 2,4 GHz)	700 €	10 %	6 meses	60 meses	7 €
			TOTAL		**210.2 €**

Tabla 24 – Costes ordenadores

- Software: Se han necesitado licencias para los programas que se especifican a continuación. La amortización se ha calculado con la Ecuación 18.

- *Matlab R2011b de MatchWorks*: Utilizado para la implementación del sistema, cuya licencia tiene un coste de 2000 €.
- *Office para Mac Hogar y Estudiantes 2011*: Empleado para la redacción de la documentación, cuya licencia tiene un coste de 119 €.

Software	Coste (sin IVA)	Uso dedicado al proyecto	Dedicación	Periodo de depreciación	Coste imputable
Matlab R2011b de MatchWorks	2000 €	100 %	6 meses	36 meses	33.3 €
Office para Mac Hogar y Estudiantes	119 €	100 %	6 meses	36 meses	19.83 €
TOTAL					**53.13 €**

Tabla 25 – Costes software

- Vídeos de prueba: Cada uno de los vídeos que se han utilizado para las pruebas tiene un coste asociado de 0.72 €. Se cuenta con 5 vídeos de prueba.

En la

Tabla 26 se resumen los costes materiales.

Descripción	Coste
Ordenadores	210.2 €
Software	53.13 €
Batería de pruebas	3.6 €
TOTAL	**266.93 €**

Tabla 26 – Costes materiales

6.4. COSTES INDIRECTOS

Los costes indirectos se calculan a priori considerando un porcentaje estimando del 20% de los costes directos (costes personales y costes materiales).

El total de costes indirectos asciende a 3781.38 €.

6.5. CUADRO RESUMEN DEL PRESUPUESTO

PRESUPUESTO

Costes directos	18906.93 €
Costes indirectos	3781.38 €
Total sin I.V.A.	22688.31 €
I.V.A (21%)	4764.54 €
Total con I.V.A	**27452.85 €**

Tabla 27 – Resumen de presupuesto

El presupuesto total del trabajo asciende a la cuantía de VEINTISIETE MIL CUATROCIENTOS CINCUENTA Y DOS EUROS CON OCHENTA Y CINCO CÉNTIMOS DE EURO.

Bibliografía

[1] “El método de contornos activos Snake tradicional”. Facultad de ingeniería de la Universidad Nacional Autónoma de México.

[2] “HIPR2: Image Processing Learning Resources” Universidad de Edimburgo, Escocia.

[3] “Procesado de imágenes en el dominio de la frecuencia”. Departamento de electrónica. Universidad de Alcalá, España.

[4] Blanco Iturralde, David Roberto; Chávez Sánchez, Juan Daniel. “Sistema de reconocimiento facial utilizando el análisis de componentes principales con una red neuronal backpropagation desarrollada en C# y Matlab”. Universidad Politécnica Salesiana de Quito, Ecuador.

[5] De la Calle Silos, Fernando. “Detección de eventos en secuencias con multitudes”. Universidad Carlos III de Madrid, España.

[6] De Miguel Benito, Darío. “Detección Automática del Color de la piel en imágenes bidimensionales basado en el análisis de regiones”. Universidad Rey Juan Carlos, España.

[7] Fawcett, Tom; “An introduction to roc analysis” .

[8] Fuentes, Luis M; Velastin, Sergio A. “People tracking in survillance aplications”. Departament of Electronic Engineering. Universidad de Londres, Reino Unido.

[9] Gámez Jiménez, Carmen Virginia. “Diseño y desarrollo de un sistema de reconocimiento de caras”. Universidad Carlos III de Madrid, España.

[10] González, Albano. “Realce y Restauración de imágenes”. Departamento de Física Fundamental y Experimental. Universidad de La Laguna, España.

[11] González, R.C. “Digital Image Processing using MATLAB”

[12] González, R.C. “Digital Image Processing”.

[13] Herrero Vez, Tamara. “Sistema automático de detección y etiquetado de caras en imágenes”. Universidad Carlos III de Madrid, España.

[14] Intel Developer. “Color Models”.

[15] Izquierdo Guerra, Walker; García Reyes, Edel. “Seguimiento y conteo de personas en ambientes exteriores con una cámara fija”. Centro de Aplicaciones de Tecnologías Avanzada. Ciudad de La Habana, Cuba.

[16] Leiva Murillo, José Miguel; Díaz de María, Fernando. Diapositivas de la asignatura “Tratamiento Digital de la Imagen”. Universidad Carlos III de Madrid, España.

[17] Marcial Basilio, Jorge A.; Aguilar Torres, Gualberto; Sánchez Pérez, Gabriel; Medina Toscano, Karina; Pérez Meana, Héctor M. “Novedosa técnica para la detección de imágenes pornográficas empleando modelos de color HSV y YCbCr”. Escuela Superior de Ingeniería Mecánica y Eléctrica, Unidad Culhuacan, México DF.

[18] Marcial Basilio, Jorge A.; Sánchez Pérez, Gabriel; Aguilar Torres, Gualberto. “Detección de imágenes con contenido explícito usando los modelos de color HSV y YCbCr”. Escuela Superior de Ingeniería Mecánica y Eléctrica, Unidad Culhuacan, México DF.

[19] Microsoft Developer Network. “Color Conversion”

[20] Muñoz Pérez, José. “Segmentación”. Universidad de Málaga, España.

[21] Nicolás Pina, Antonio. “Clasificación y búsqueda de imágenes usando característica visuales”. Universidad de Murcia, España.

[22] Organización de soporte técnico de Microsoft.

[23] Pérez, C; Vicente, A; Fernández, C; Reinoso, O; Gil, A. “Aplicación de los diferentes espacios de color para la detección y el seguimiento de caras”. Departamento de Ingeniería de Sistemas Industriales. Universidad Miguel Hernández de Alicante, España.

[24] Pinho, Raquel; Tavares João, Manuel; Correia Miguel. “An Efficient and Robust Tracking System using Kalman Filter”. Universidad de Oporto, Portugal.

[25] Prem Kuchi; Prasad Gabbur; P. Subbanna Bhat; Suman David. “Human Face Detection and Tracking using Sking Color Modeling and Connected Component Operators”. IETE Journal of research,2002.

[26] Rius, Ignasi; Rowe, Daniel; González, Jordi; Roca, Xavier. “A 3D Dynamic Model of Human Actions for Probabilistic Image Tracking”. Centro de Visión por Ordenador. Universidad Autónoma de Barcelona, España.

[27] Rius, Ignasi; Varona, Javier; González, Jord; Villanueva, Juan J. “Action Spaces for Efficent Bayesian Tranking of Human Motion”. Centro de Visión por Ordenador. Universidad Autónoma de Barcelona, España.

[28] Sánchez de la Rosa, José Luis. “Matlab / Octave”. Universidad de La Laguna, España.

[29] Terrillon, J. -C.; Shirazi, M. N.; Fukamachi, H.; Akamatsu, S.“Comparative performance of different skin chrominance models and chrominance spaces for the automatic detection of human faces in color images”. Automatic Face and Gesture Recognition, Proceedings Fourth IEEE International Conference, 2000.

[30] Traba Martínez, Lola; Curras Martínez, Manuel. “Detección de bordes”.

[31] Yánez García, Javier. “Tracking de personas a partir de visión artificial”. Universidad Carlos II de Madrid, España.

[32] Zarit, B. D.; Super, B. J. and Quek, F. K. H. “Comparison of five color models in skin pixel classification”.